

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP04/019102

International filing date: 21 December 2004 (21.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2003-424042  
Filing date: 22 December 2003 (22.12.2003)

Date of receipt at the International Bureau: 24 February 2005 (24.02.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日本国特許庁  
JAPAN PATENT OFFICE

24.12.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2003年12月22日

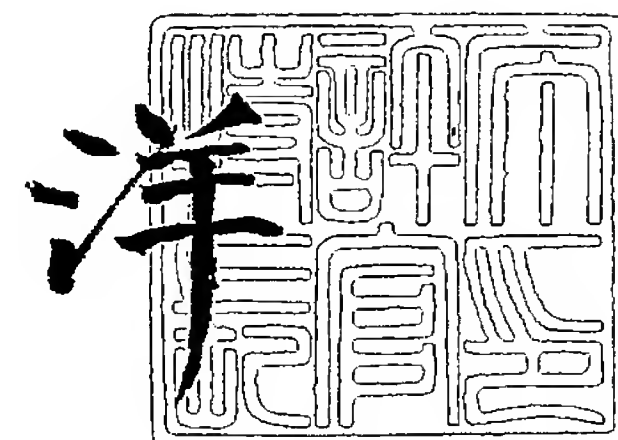
出願番号  
Application Number: 特願2003-424042  
[ST. 10/C]: [JP2003-424042]

出願人  
Applicant(s): 松下電器産業株式会社

2005年 2月10日

特許庁長官  
Commissioner,  
Japan Patent Office

小川



【書類名】 特許願  
【整理番号】 2037950003  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 12/08 310  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 中西 龍太  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 岡林 はづき  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 田中 哲也  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 宮阪 修二  
【特許出願人】  
    【識別番号】 000005821  
    【氏名又は名称】 松下電器産業株式会社  
【代理人】  
    【識別番号】 100109210  
    【弁理士】  
    【氏名又は名称】 新居 広守  
【手数料の表示】  
    【予納台帳番号】 049515  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 0213583

**【書類名】 特許請求の範囲****【請求項 1】**

キャッシュの単位データを保持するキャッシュエントリーに対応させて、当該キャッシュエントリーが有効であるか否かを示すバリッドフラグと、当該キャッシュエントリーに対する書き込みがなされたか否かを示すダーティフラグとを保持するフラグ保持手段と、プロセッサからの指示に基づき、前記バリッドフラグおよびダーティフラグの少なくとも一方をキャッシュエントリーの状態に反して改変する改変手段とを備えることを特徴とするキャッシュメモリ。

**【請求項 2】**

前記改変手段は、メモリからデータをロードすることなく、キャッシュエントリーに対してタグとしてのアドレスを設定しバリッドフラグをセットすることを特徴とする請求 1 記載のキャッシュメモリ。

**【請求項 3】**

前記改変手段は、キャッシュエントリーに対してライトバックされていない書き換えられたデータを保持している状態でダーティフラグをリセットすることを特徴とする請求項 2 記載のキャッシュメモリ。

**【請求項 4】**

前記キャッシュメモリは、さらに、プロセッサから指定されたアドレス範囲を保持する保持手段と、保持されたアドレス範囲に属するデータを保持するキャッシュエントリーを特定する特定手段とを備え、前記改変手段は、特定されたキャッシュエントリーに対して前記バリッドフラグおよびダーティフラグの少なくとも一方を改変することを特徴とする請求項 2 又は 3 記載のキャッシュメモリ。

**【請求項 5】**

前記特定手段は、前記アドレス範囲の先頭アドレスがラインデータの途中を指す場合、当該先頭アドレスを、前記アドレス範囲に含まれる先頭のラインを指すスタートラインアドレスに変換する第 1 変換手段と、前記アドレス範囲の末尾アドレスがラインデータの途中を指す場合、当該末尾アドレスを、前記アドレス範囲に含まれる末尾のラインを指すエンドラインアドレスに変換する第 2 変換手段と、前記スタートラインアドレスからエンドラインアドレスまでの各ラインアドレスに対応するデータを保持するキャッシュエントリーがあるか否かを判定する判定手段とを備えることを特徴とする請求項 4 記載のキャッシュメモリ。

**【請求項 6】**

前記改変手段は、ダーティフラグのリセット指示付きメモリアクセス命令が実行されたことを検出する命令検出手段と、当該命令によってアクセスがなされたキャッシュエントリーに対してダーティフラグをリセットするフラグ書き換え手段とを備えることを特徴とする請求項 1 記載のキャッシュメモリ。

**【請求項 7】**

前記改変手段は、バリッドフラグのリセット指示付きメモリアクセス命令が実行されたことを検出する命令検出手段と、当該命令によってアクセスがなされたキャッシュエントリーに対してバリッドフラグをリセットするフラグ書き換え手段とを備えることを特徴とする請求項 1 記載のキャッシュメモリ。

**【請求項 8】**

キャッシュの単位データを保持するキャッシュエントリーに対応させて、当該キャッシュエントリーが有効であるか否かを示すバリッドフラグと、当該キャッシュエントリーに対する書き込みがなされたか否かを示すダーティフラグとを有するキャッシュメモリの制御方法であって、

プロセッサの指示に基づき、メモリからデータをロードすることなく、キャッシュエントリーに対してタグとしてのアドレスを設定しバリッドフラグをセットするステップと、

プロセッサの指示に基づき、キャッシュエントリーに対してライトバックされていない書き換えられたデータを保持している状態でダーティフラグをリセットするステップとを有することを特徴とする制御方法。

【書類名】 明細書

【発明の名称】 キャッシュメモリ及びその制御方法

【技術分野】

【0 0 0 1】

本発明は、プロセッサのメモリアクセスを高速化するためのキャッシュメモリおよびその制御方法に関する。

【背景技術】

【0 0 0 2】

近年のマイクロプロセッサでは、例えば、SRAM (Static Random Access Memory) 等から成る小容量で高速なキャッシュメモリをマイクロプロセッサの内部、もしくはその近傍に配置し、データの一部をキャッシュメモリに記憶することによって、マイクロプロセッサのメモリアクセスを高速化させている。

コンピュータシステムでは、中央処理装置からキャッシュメモリへのリードアクセスまたはライトアクセスがミスヒットした場合に、主記憶装置から新たに読み出されたデータの一部が、エントリ（登録項目）としてキャッシュメモリの空きブロックに格納される。この時、空きブロックが存在しない場合には、複数のブロックのいずれか1つを選択し、選択されたブロックに格納されているエントリを主記憶装置に戻してブロック内を空き状態にし、この空きブロックに新たに読み出したデータを格納するエントリ置換処理が必要になる。上記エントリ置換処理では、最も以前に参照したデータを格納しているブロックを選択する手法、すなわち、LRU (Least Recently Used) デコード方式が一般的に採用されている。このLRUデコード方式によってキャッシュメモリの使用効率が向上し、その結果、マイクロプロセッサの実行速度が向上する。

【0 0 0 3】

マイクロプロセッサが処理するプログラムの中には、アクセス頻度は少ないものの、ひとたび起動された場合には高速に処理しなければならないような特殊な処理と、アクセス頻度は多いが、実行速度がそれほど要求されないような処理とが存在する。

そこで、これに対応するために例えば特許文献1等の従来技術では、キャッシュメモリにフリーズ機能を設けている。フリーズ機能は、アクセスは少ないものの、ひとたび起動された場合には高速に処理しなければならないようなプログラムを予めキャッシュメモリ内にコピーしておき、その領域を書き換え禁止にしておく機能である。この機能を有することで、コンピュータシステムは、必要な時にプログラムをキャッシュメモリから読み出して実行することができ、これにより実行時間が短縮する。また、パージ機能は、アクセス頻度は多いが、実行速度がそれほど要求されないようなプログラムやデータをキャッシュメモリ内に保存しておくことなく、その領域を解放する機能である。この機能を有することで、キャッシュメモリに余裕ができ、優先度の高い他のプログラムやデータをキャッシュメモリに取り込むことができ、これにより、キャッシュメモリの利用効率が向上し、総合的な実行時間が短縮する。

【特許文献1】 特開 2 0 0 0 - 2 0 0 2 2 1 号公報

【発明の開示】

【発明が解決しようとする課題】

【0 0 0 4】

しかしながら、フリーズ機能を使用してもパージ機能を使用しても、キャッシュミス発生によりリプレースする場合に無駄なリプレースをしたり、無駄なライトバックをする場合があるという問題がある。

例えば、無駄なリプレースが生じる場合として、プロセッサが配列要素を全てライトする場合がある。この場合、メモリから新たにデータをキャッシュメモリにリプレースしても、プロセッサから全てライトされるので、リプレースが無駄に終わる。

【0 0 0 5】

また、無駄なライトバックが生じる場合として、キャッシュエントリーが単にワークデータとして使用される場合がある。この場合、最終的に破棄してもよいデータであるにも



拘らず、無駄なライトバックが発生する。

本発明は、無駄なリプレイスや無駄なライトバックを防止するキャッシュメモリを提供することを目的とする。

【課題を解決するための手段】

【0006】

上記課題を解決するため本発明のキャッシュメモリは、キャッシュの単位データを保持するキャッシュエントリーに対応させて、当該キャッシュエントリーが有効であるか否かを示すバリッドフラグと、当該キャッシュエントリーに対する書き込みがなされたか否かを示すダーティフラグとを保持するフラグ保持手段と、プロセッサから指示に基づき、前記バリッドフラグおよびダーティフラグの少なくとも一方をキャッシュエントリーの状態に反して改変する改変手段とを備える。

【0007】

前記改変手段は、キャッシュエントリーに対してタグとしてのアドレスを設定し、有効なデータを保持していない状態でバリッドフラグをセットする構成としてもよい。

この構成によれば、配列などのデータを書き込むためのキャッシュエントリーをキャッシュメモリ上に確保することができ、しかも、上書きされる運命にあるデータをメモリからキャッシュメモリへ無駄にロードすることを防止することができる。

【0008】

また、前記改変手段は、プロセッサからの指定されたキャッシュエントリーに対してライトバックされていない書き換えられたデータを保持している状態でダーティフラグをリセットする。

この構成によれば、テンポラリーなワークデータ等破棄される運命にあるデータを保持するキャッシュエントリーから無駄にライトバックすることを防止することができる。

【0009】

ここで、前記キャッシュメモリは、さらに、プロセッサから指定されたアドレス範囲を保持する保持手段と、保持されたアドレス範囲に属するデータを保持するキャッシュエントリーを特定する特定手段とを備え、前記改変手段は、特定されたキャッシュエントリーに対して前記バリッドフラグおよびダーティフラグの少なくとも一方を改変する構成としてもよい。

この構成によれば、プロセッサから指定されたアドレス範囲で、キャッシュメモリへの無駄なデータのロード又はメモリへの無駄なライトバックを防止することができる。

【0010】

ここで、前記特定手段は、前記アドレス範囲の先頭アドレスがラインデータの途中を指す場合、当該先頭アドレスを、前記アドレス範囲に含まれる先頭のラインを指すスタートラインアドレスに変換する第1変換手段と、前記アドレス範囲の末尾アドレスがラインデータの途中を指す場合、当該末尾アドレスを、前記アドレス範囲に含まれる末尾のラインを指すエンドラインアドレスに変換する第2変換手段と、前記スタートラインアドレスからエンドラインアドレスまでの各ラインアドレスに対応するデータを保持するキャッシュエントリーがあるか否かを判定する判定手段とを備える構成としてもよい。

【0011】

この構成によれば、プロセッサは、前記アドレス範囲として、キャッシュメモリのラインサイズ及びライン境界のアドレスとは無関係に任意のアドレスから任意のアドレス（又は任意のサイズ）を指定することができる。つまり、プロセッサにおいてキャッシュメモリのラインサイズ及びライン境界のアドレスを管理する必要がないので、キャッシュメモリ管理のための負荷を解消することができる。

【0012】

ここで、前記改変手段は、ダーティフラグのリセット指示付きメモリアクセス命令が実行されたことを検出する命令検出手段と、当該命令によってアクセスがなされたキャッシュエントリーに対してダーティフラグをリセットするフラグ書き換え手段とを備える構成としてもよい。

## 【0013】

ここで、前記改変手段は、バリッドフラグのリセット指示付きメモリアクセス命令が実行されたことを検出する命令検出手段と、当該命令によってアクセスがなされたキャッシュエントリーに対してバリッドフラグをリセットするフラグ書き換え手段とを備える構成としてもよい。

また、本発明のキャッシュメモリの制御方法についても上記と同様の手段、作用を有する。

## 【発明の効果】

## 【0014】

本発明のキャッシュメモリによれば、配列などのデータを書き込むためのキャッシュエントリーをキャッシュメモリ上に確保することができ、しかも、上書きされる運命にあるデータをメモリからキャッシュメモリへ無駄にロードすることを防止することができる。

また、テンポラリーなワークデータ等破棄される運命にあるデータを保持するキャッシュエントリーから無駄にライトバックすることを防止することができる。

## 【0015】

さらに、プロセッサは、前記アドレス範囲として、キャッシュメモリのラインサイズ及びライン境界のアドレスとは無関係に任意のアドレスから任意のアドレス（又は任意のサイズ）を指定することができる。つまり、プロセッサにおいてキャッシュメモリのラインサイズ及びライン境界のアドレスを管理する必要がないので、キャッシュメモリ管理のための負荷を解消することができる。

## 【発明を実施するための最良の形態】

## 【0016】

（実施の形態1）

## &lt;全体構成&gt;

図1は、本発明の実施の形態1におけるプロセッサ1、キャッシュメモリ3、メモリ2を含むシステムの概略構成を示すブロック図である。同図のように、本発明のキャッシュメモリ3は、プロセッサ1およびメモリ2を有するシステムに備えられる。

## 【0017】

キャッシュメモリ3は、キャッシュエントリー毎にキャッシュエントリーが有効であるか否かを示すバリッドフラグVと、当該キャッシュエントリーに対して書き込みがなされたか否かを示すダーティフラグDとを有し、プロセッサ1により指定されたアドレスのデータを保持するキャッシュエントリーに対してデータをライトバックすることなくダーティフラグDを1から0（ダーティでない）に改変し、また、プロセッサ1により指定されたアドレスに対応するキャッシュエントリーを確保してメモリからデータをロードすることなくバリッドフラグVを1（有効）に改変するよう構成されている。

## 【0018】

ダーティフラグDを1から0に改変するのは、最終的には破棄されるテンポラリーなワークデータを保持するキャッシュエントリーから無駄にライトバックすることを防止するためである。また、キャッシュエントリーを確保してメモリからデータをロードすることなくバリッドフラグVを1（有効）に改変することにより、配列など書き込みをするためのキャッシュエントリーを予め確保するためである。

## 【0019】

## &lt;キャッシュメモリの構成&gt;

以下、キャッシュメモリ3の具体例として、4ウェイ・セット・アソシエイティブ方式のキャッシュメモリに本発明を適用した場合の構成について説明する。

## 【0020】

図2は、キャッシュメモリ3の構成例を示すブロック図である。同図のように、キャッシュメモリ3は、アドレスレジスタ20、メモリI/F21、デコーダ30、4つのウェイ31a～31d（以下ウェイ0～3と略す）、4つの比較器32a～32d、4つのアンド回路33a～33d、オア回路34、セクタ35、36、デマルチプレクサ37、



制御部 3 8 を備える。

【0 0 2 1】

アドレスレジスタ 2 0 は、メモリ 2 へのアクセスアドレスを保持するレジスタである。このアクセスアドレスは 3 2 ビットであるものとする。同図に示すように、アクセスアドレスは、最上位ビットから順に、2 1 ビットのタグアドレス、4 ビットのセットインデックス（図中の S I）、5 ビットのワードインデックス（図中の W I）を含む。ここで、タグアドレスはウェイにマッピングされるメモリ中の領域（そのサイズはセット数×ブロックである）を指す。この領域のサイズは、タグアドレスよりも下位のアドレスビット（A 1 0 ~ A 0）で定まるサイズつまり 2 k バイトであり、1 つのウェイのサイズでもある。セットインデックス（S I）はウェイ 0 ~ 3 に跨る複数セットの 1 つを指す。このセット数は、セットインデックスが 4 ビットなので 1 6 セットある。タグアドレスおよびセットインデックスで特定されるキャッシュエントリは、リプレース単位であり、キャッシュメモリに格納されている場合はラインデータ又はラインと呼ばれる。ラインデータのサイズは、セットインデックスよりも下位のアドレスビットで定まるサイズつまり 1 2 8 バイトである。1 ワードを 4 バイトとすると、1 ラインデータは 3 2 ワードである。ワードインデックス（W I）は、ラインデータを構成する複数ワード中の 1 ワードを指す。アドレスレジスタ 2 0 中の最下位 2 ビット（A 1、A 0）は、ワードアクセス時には無視される。

【0 0 2 2】

メモリ I / F 2 1 は、キャッシュメモリ 3 からメモリ 2 へのデータのライトバックや、メモリ 2 からキャッシュメモリ 3 へのデータのロード等、キャッシュメモリ 3 からメモリ 2 をアクセスするための I / F である。

デコーダ 3 0 は、セットインデックスの 4 ビットをデコードし、4 つのウェイ 0 ~ 3 に跨る 1 6 セット中の 1 つを選択する。

4 つのウェイ 0 ~ 3 は、同じ構成を有数する 4 つのウェイであり、4 × 2 k バイトの容量を有する。各ウェイは、1 6 個のキャッシュエントリを有する。

【0 0 2 3】

図 3 に 1 つのキャッシュエントリにおける詳細なビット構成を示す。同図のように、1 つのキャッシュエントリは、バリッドフラグ V 0 ~ V 3、2 1 ビットのタグ、1 2 8 バイトのラインデータ、ダーティフラグ D 0 ~ D 3 を有する。

【0 0 2 4】

タグは 2 1 ビットのタグアドレスのコピーである。

ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の 1 2 8 バイトデータのコピーであり、3 2 バイトの 4 つのサブラインからなる。

バリッドフラグ V 0 ~ V 3 は、4 つのサブラインに対応し、サブラインが有効か否かを示す。

【0 0 2 5】

ダーティフラグ D 0 ~ D 3 は、4 つのサブラインに対応し、そのサブラインにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。

比較器 3 2 a は、アドレスレジスタ 2 0 中のタグアドレスと、セットインデックスにより選択されたセットに含まれる 4 つのタグ中のウェイ 0 のタグとが一致するか否かを比較する。比較器 3 2 b ~ 3 2 c についても、ウェイ 3 1 b ~ 3 1 d に対応すること以外は同様である。

【0 0 2 6】

アンド回路 3 3 a は、バリッドフラグと比較器 3 2 a の比較結果とが一致するか否かを比較する。この比較結果を h 0 とする。比較結果 h 0 が 1 である場合は、アドレスレジスタ 2 0 中のタグアドレスおよびセットインデックスに対応するラインデータが存在すること、つまりウェイ 0 においてヒットしたことを意味する。比較結果 h 0 が 0 である場合は

、ミスヒットしたことを意味する。アンド回路 3 3 b ~ 3 3 d についても、ウェイ 3 1 b ~ 3 1 d に対応すること以外は同様である。その比較結果 h 1 ~ h 3 は、ウェイ 1 ~ 3 でヒットしたかミスしたかを意味する。

#### 【0 0 2 7】

オア回路 3 4 は、比較結果 h 0 ~ h 3 のオアをとる。このオアの結果を h i t とする。h i t は、キャッシュメモリにヒットしたか否かを示す。

セクタ 3 5 は、選択されたセットにおけるウェイ 0 ~ 3 のラインデータのうち、ヒットしたウェイのラインデータを選択する。

セクタ 3 6 は、セクタ 3 5 により選択された 3 2 ワードのラインデータのうち、ワードインデックスに示される 1 ワードを選択する。

#### 【0 0 2 8】

デマルチプレクサ 3 7 は、キャッシュエントリにデータを書き込む際に、ウェイ 0 ~ 3 の 1 つに書き込みデータを出力する。この書き込みデータはワード単位でよい。

制御部 3 8 は、キャッシュメモリ 3 の全体の制御を行う。特に、プロセッサからのコマンド及びアドレス指定に従って、V フラグの改変と、D フラグの改変とを行う。

#### 【0 0 2 9】

##### <制御部の構成>

図 4 は、制御部 3 8 の構成を示すブロック図である。同図のように、制御部 3 8 は、フラグ更新部 3 9、リプレース部 4 0、フラグ改変部 4 1 とを含む。

#### 【0 0 3 0】

フラグ更新部 3 9 は、従来技術と同様にキャッシュメモリの状態を反映するよう V フラグ、D フラグの更新を行う。

リプレース部 4 0 は、従来技術と同様にキャッシュエントリーのリプレースを行う。

フラグ改変部 4 1 は、プロセッサ 1 からのコマンド及びアドレス指定に応じて、V フラグの改変と D フラグの改変とを行う。このコマンドには、V フラグ設定コマンドと D フラグ設定コマンドとがある。

#### 【0 0 3 1】

##### <フラグ改変部の構成>

図 5 は、フラグ改変部 4 1 の構成例を示すブロック図である。同図のようにフラグ改変部 4 1 は、コマンドレジスタ 4 0 1、スタートアドレスレジスタ 4 0 2、サイズレジスタ 4 0 3、加算器 4 0 4、スタートアライナ 4 0 5 a、4 0 5 b、エンドアライナ 4 0 6 a、4 0 6 b、フラグ書換部 4 0 7 を備える。

#### 【0 0 3 2】

コマンドレジスタ 4 0 1 は、プロセッサ 1 から直接アクセス可能なレジスタであり、プロセッサ 1 により書き込まれたコマンドを保持する。図 6 (c) に、コマンドレジスタ 4 0 1 にコマンドを書き込む命令の一例を示す。この命令は、通常の転送命令 (m o v 命令) であり、ソースオペランドとしてコマンドを、デスティネーションオペランドとしてコマンドレジスタ (C R) 4 0 1 を指定している。図 6 (d) に、コマンドフォーマットの一例を示す。このコマンドフォーマットは、V フラグ設定コマンド用の 2 ビットのフィールドと、D フラグ設定コマンド用の 2 ビットのフィールドと、サブライン指定用の 4 ビットのフィールドとを含む。

#### 【0 0 3 3】

例えば、V フラグ設定コマンドは、「1 0」であれば V = 0 に設定することを指示し (V リセットコマンド)、「1 1」であれば V = 1 に設定することを指示し (V セットコマンド)、「0 0」であれば無効 (ノーオペレーション) を意味する。D フラグ設定コマンドも同様である。サブライン指定フィールドは、サブラインを個別に指定するためのフィールドであり、個別指定がない場合はライン (全サブライン) 指定であるものとする。

#### 【0 0 3 4】

スタートアドレスレジスタ 4 0 2 は、プロセッサ 1 から直接アクセス可能なレジスタであり、プロセッサ 1 により書き込まれたスタートアドレスを保持する。このスタートアド

レスはCフラグを設定すべきアドレス範囲の開始位置を示す。図6(a)に、スタートアドレスレジスタ402にスタートアドレスを書き込む命令の一例を示す。この命令も、図6(c)と同様に通常の転送命令(mov命令)である。

#### 【0035】

サイズレジスタ403は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたサイズを保持する。このサイズは、スタートアドレスからのアドレス範囲を示す。図6(b)に、サイズレジスタ403にサイズを書き込む命令の一例を示す。この命令も、図6(c)と同様に通常の転送命令(mov命令)である。なお、サイズの単位は、バイト数であっても、ライン数(キャッシュエントリー数)であってもよく、予め定められた単位であればよい。

#### 【0036】

加算器404は、スタートアドレスレジスタ402に保持されたスタートアドレスとサイズレジスタ403に保持されたサイズとを加算する。加算結果は、アドレス範囲の終了位置を指すエンドアドレスである。加算器404は、サイズがバイト数指定の場合はバイトアドレスとして加算し、サイズがライン数指定の場合はラインアドレスとして加算すればよい。

#### 【0037】

スタートアライナ405a、405bは、スタートアドレスをライン境界の位置に調整する。スタートアライナ405aはエンドアドレスの方向に、405bはエンドアドレスとは反対の方向に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意のアドレスをスタートアドレスとして指定することができる。

エンドアライナ406a、406bは、エンドアドレスをライン境界の位置に調整する。エンドアライナ406aはスタートアドレスの方向に、406bはスタートアドレスとは反対の方向に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意の大きさを上記サイズとして指定することができる。

#### 【0038】

図7に、スタートアライナ405a、405b及びエンドアライナ406a、406bの説明図を示す。同図において、プロセッサ1から指定されたスタートアドレスはラインNの途中の任意の位置を指す。スタートアライナ405aは、次のライン(N+1)の先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスaとして出力する。スタートアライナ405bは、スタートアドレスのデータを含むラインNの先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスbとして出力する。アラインスタートアドレスが指すラインをスタートラインと呼ぶ。

#### 【0039】

また、エンドアドレスはラインMの途中の任意の位置を指す。エンドアライナ406aは、直前のライン(M-1)の先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスaとして出力する。エンドアライナ406bは、エンドアドレスのデータを含むラインMの先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスbとして出力する。アラインエンドアドレスが指すラインをエンドラインと呼ぶ。

#### 【0040】

同図のように、スタートアライナ405a及びエンドアライナ406aはライン単位で内側アラインを行う。スタートアライナ405b及びエンドアライナ406bはライン単位で外側アラインを行う。さらに、ライン単位の外側アラインの後、さらに、サブライン単位の外側アラインと内側アラインが可能である。

フラグ書換部407は、スタートラインからエンドラインまで、コマンドに従ってVフラグ又はDフラグの値を設定する。その際、スタートライン及びエンドラインが内側アラインか外側アラインかはコマンドに応じて選択される。

#### 【0041】

<フラグ改変処理>

図8は、フラグ書換部407におけるフラグ改変処理の一例を示すフローチャートであ



る。同図では、DリセットコマンドとVセットコマンドとを示している。

#### 【0042】

同図において、コマンドレジスタ401にDリセットコマンドが書き込まれている場合（S80）、フラグ書換部407は、スタートアライナ405a、405b、エンドアライナ406a、406bの出力の中から内側アラインによるスタートラインとエンドラインとを選択する（S81）。ここで内側アラインを選択しているのは、ラインNとラインMにおけるスタートアドレスとエンドアドレスの外側のデータは、プロセッサ1により破棄されるとは限らないからである。

#### 【0043】

さらに、フラグ書換部407は、スタートラインからエンドラインまでの各ラインアドレスを順に出力しながらループ1の処理（S82～S86）を行う。フラグ書換部407は、各ラインについて同じ処理を行うので、ここでは1ライン分の処理について説明する。

すなわち、フラグ書換部407は、キャッシュメモリ3がプロセッサ1からアクセスされていない間に、ラインアドレスをアドレスレジスタ20に出力し（S83）、アドレスレジスタ20のタグアドレスとキャッシュエントリーのタグとを比較器32a～32dに比較させ、ヒットするかどうかを判定する（S84）。さらにフラグ書換部407は、ヒットした場合には、ヒットしたキャッシュエントリーに対してD0～D3フラグを0にリセットし（S85）、ミスヒットした場合には、キャッシュメモリにエントリーされていないのでなにもしない。

#### 【0044】

このように、スタートラインからエンドラインまでの各ラインについて、キャッシュメモリ3にエントリーされている場合には、D0～D3フラグに0が設定される。フラグ書換部407は、ループ1の終了後、コマンドレジスタ401のコマンドをクリアする（S98）。これにより、テンポラリーなワークデータ等破棄される運命にあるデータを保持するキャッシュエントリーから無駄にライトバックすることを防止することができる。

#### 【0045】

また、図8において、コマンドレジスタ401にVセットコマンドが書き込まれている場合（S87）、フラグ書換部407は、スタートアライナ405a、405b、エンドアライナ406a、406bの出力の中から外側アラインによるスタートラインとエンドラインとを選択する（S88）。ここで外側アラインを選択しているのは、内側アラインではスタートアドレスからエンドアドレスまでのサイズを確保できないからである。

#### 【0046】

さらに、フラグ書換部407は、スタートラインからエンドラインまでの各ラインアドレスを順に出力しながらループ2の処理（S89～S97）を行う。フラグ書換部407は、各ラインについて同じ処理を行うので、ここでは1ライン分の処理について説明する。

すなわち、フラグ書換部407は、キャッシュメモリ3がプロセッサ1からアクセスされていない間に、ラインアドレスをアドレスレジスタ20に出力し（S90）、アドレスレジスタ20のタグアドレスとキャッシュエントリーのタグとを比較器32a～32dに比較させ、ヒットするかどうかを判定する（S91）。さらにフラグ書換部407は、ヒットしなかった場合には、当該ラインアドレスに対応するセット内の4つのウェイからLRU方式でリプレース対象のウェイを1つ選択し（S92）、選択されたウェイのダーティフラグD0～D3の論理和が1であるか否かを判定する（S93）。この論理和が1、つまりダーティであると判定された場合、フラグ書換部407は、ダーティなサブラインのみをライトバックする（S94）。上記の論理和が1でないつまりダーティでないと判定された場合、又はダーティなサブラインのライトバックの後に、フラグ書換部407は、キャッシュエントリーにメモリからデータをロードすることなく、キャッシュエントリーにラインアドレスをタグとして設定し（S95）、バリッドフラグV0～V3を1にセッ

トする (S 9 6)。このようにして、有効なデータを保持していないが V 0 ~ V 3 = 1 と設定された 1 ライン分のキャッシュエントリーが確保される。

【0 0 4 7】

さらに、フラグ書換部 4 0 7 は、ループ 1 の終了後、コマンドレジスタ 4 0 1 のコマンドをクリアする (S 9 8)。

これにより、配列などのデータを書き込み用のキャッシュエントリーをキャッシュメモリ上に確保し、かつ、無駄なライトバックを防止することができる。

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、D リセットコマンドにより、破棄される運命にあるデータを保持するキャッシュエントリーから無駄にライトバックすることを防止することができる。また、V セットコマンドにより、配列などのデータを書き込み用のキャッシュエントリーをキャッシュメモリ上に確保し、かつ、無駄なライトバックを防止することができる。

【0 0 4 8】

なお、上記実施の形態では、D リセットコマンドと V セットコマンドについて説明したが、V リセットコマンドについては、図 8 に示した S 8 7 ~ S 9 8 において、ステップ S 9 5 を削除し、ステップ S 9 6 において V 0 ~ V 3 を 0 にリセットすることにより実現することができる。これによれば、無駄なライトバックを防止してキャッシュエントリーを開放することができる。

【0 0 4 9】

また、D セットコマンドは、図 8 の S 8 0 ~ S 8 6 と同様に実行可能ではあるが、プロセッサ 1 からストア命令によりデータを書き込めば D フラグがセットされること、D フラグのセットによりライトバック動作が発生することを考えれば、あまり実用的とはいえない。しかし、キャッシュメモリのテスト動作や性能の評価や検証等に利用することができる。

【0 0 5 0】

<変形例>

なお、本発明のキャッシュメモリは、上記の実施の形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 上記実施の形態では、V 0 ~ V 3 フラグ、D 0 ~ D 3 フラグを同時にセット又はリセットしているが、サブ単位にセット又はリセットするようにしてもよい。

【0 0 5 1】

例えば、スタートラインとエンドラインのみをサブライン単位で処理する場合、フラグ書換部 4 0 7 は、外側アライン (ライン) されたスタートアドレスとエンドアドレスを選択し、さらに外側アライン (サブライン) または内側アライン (サブライン) によりスタートラインのサブラインアドレスと、エンドラインのサブラインアドレスとを算出し、スタートラインとエンドラインのみをサブライン単位で処理すればよい。また、プロセッサ 1 は、コマンド中のサブライン指定フィールドにおいてその旨を指定すればよい。

【0 0 5 2】

例えば、プロセッサ 1 がサイズレジスタ 4 0 3 のサイズを 0 に指定し、コマンド中のサブライン指定フィールドにおいて特定のサブラインを指定した場合に、フラグ書換部 4 0 7 は、指定されたサブラインのみを対象に処理をすればよい。

【0 0 5 3】

(2) また、フラグ改変部 4 1 は、ダーティフラグのリセット指示付きメモリアクセス命令が実行されたこと検出する命令検出部と、当該命令によってアクセスがなされたキャッシュエントリーに対してダーティフラグをリセットするフラグ書き換え手部とを備える構成としてもよい。

さらに、命令検出部は、バリッドフラグのリセット指示付きメモリアクセス命令が実行されたこと検出し、フラグ書き換え手部は、当該命令によってアクセスがなされたキャッシュエントリーに対してバリッドフラグをリセットする構成としてもよい。

【0 0 5 4】



(3) 上記実施の形態では、4 ウェイ・セット・アソシエイティブのキャッシュメモリを例に説明したが、ウェイ数は、いくつでもよい。また、上記実施の形態では、セット数が 16 である例を説明したが、セット数はいくつでもよい。

(4) 上記実施の形態では、セット・アソシエイティブのキャッシュメモリを例に説明したが、フル・アソシエイティブ方式のキャッシュメモリであってもよい。

(5) 上記実施の形態では、サブラインのサイズをラインのサイズの  $1/4$  としているが、 $1/2$ 、 $1/8$ 、 $1/16$  等他のサイズでもよい。その場合、各キャッシュエントリーは、サブラインと同数のバリッドフラグおよびダーティフラグをそれぞれ保持すればよい。

#### 【0055】

##### (実施の形態 2)

実施の形態 1 では、V フラグの改変と D フラグの改変とを行う構成について説明したが、本実施の形態では、これ以上使用（書き込み及び読み出し）するかどうかを示す W（ウィーク）フラグを有するキャッシュメモリにおいて W フラグを改変する構成について説明する。

#### 【0056】

##### <キャッシュメモリの構成>

図 9 は、本発明の実施の形態 2 におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図 2 の構成と比較して、ウェイ 31a～31d の代わりにウェイ 131a～131d を備える点と、制御部 38 の代わりに制御部 138 を備える点とが異なっている。以下、同じ点は説明を省略して、異なる点を中心に説明する。

#### 【0057】

ウェイ 131a は、ウェイ 31a と比べて、各キャッシュエントリー中に W フラグ及び U フラグが追加されている点異なる。ウェイ 131b～131d も同様である。

図 10 に、キャッシュエントリーのビット構成を示す。1 つのキャッシュエントリーは、バリッドフラグ V0～V3、21 ビットのタグ、128 バイトのラインデータ、ウィークフラグ W、使用フラグ U 及びダーティフラグ D0～D3 を保持する。

#### 【0058】

このうち、ウィークフラグ W は、プロセッサからのアクセスに関しては、これ以上使用するかどうかを意味し、キャッシュメモリにおけるリプレース制御に関しては、他のキャッシュエントリーよりも真っ先に追い出してもよい最弱のリプレース対象を意味する。このように、ウィークフラグ W は二つの意味を有することから、クリーニング処理とリプレース処理との 2 つの処理で参照される。

#### 【0059】

使用フラグ U は、そのキャッシュエントリーにアクセスがあったか否かを示し、LRU 方式におけるミスヒットによるリプレースに際して 4 つのウェイのキャッシュエントリー間におけるアクセス順序データの代わりに用いられる。より正確には、使用フラグ U の 1 は、アクセスがあったことを、0 はないことを意味する。ただし、1 つのセット内の 4 つウェイの使用フラグが全て 1 になれば、0 にリセットされる。別言すれば、使用フラグ U は、アクセスされた時期が古いか新しいか 2 つの相対的な状態を示す。つまり、使用フラグ U が 1 のキャッシュエントリーは、使用フラグが 0 のキャッシュエントリーよりも新しくアクセスされたことを意味する。

制御部 138 は、制御部 38 と比べて、W フラグを設定する点と、LRU 方式におけるアクセス順序情報の代わりに使用フラグ U を用いる点とが異なる。

#### 【0060】

##### <制御部の構成>

図 11 は、制御部 138 の構成を示すブロック図である。同図の制御部 138 は、制御部 38 と比較して、フラグ更新部 39、リプレース部 40、フラグ改変部 41 の代わりにフラグ更新部 139、リプレース部 140、フラグ改変部 141 を備える点異なる。

#### 【0061】

フラグ更新部 1 3 9 は、フラグ更新部 3 9 と同様に V フラグ、D フラグを更新することに加えて、キャッシュメモリがアクセスされたときに使用フラグ U の更新処理を行う。

リプレース部 1 4 0 は、通常の L R U 方式ではなく、使用フラグ U をアクセス順序とする擬似的な L R U 方式によりリプレースを行う。ただし、リプレース処理に際して  $W=1$  のキャッシュエントリは真っ先にリプレース対象として選択する。

#### 【0 0 6 2】

フラグ改変部 1 4 1 は、プロセッサ 1 からのコマンドに応じてウィークフラグ W を設定する。プロセッサ 1 は、もはや使用（書き込み及び読み出し）をしないキャッシュエントリについてウィークフラグの設定を指示するコマンドをキャッシュメモリ 3 に対して発行する。 $W=1$  のキャッシュエントリは、キャッシュミス時には使用フラグ U の値に関わらず、真っ先にリプレース対象となる。また、 $W=1$  のキャッシュエントリがデータイであればクリーニング処理の対象となる。

#### 【0 0 6 3】

<使用フラグ U の説明>

図 1 2 は、フラグ更新部 3 9 による使用フラグ U の更新例を示す。同図の上段、中断、下段は、ウェイ 0 ～ 3 に跨るセット N を構成する 4 つのキャッシュエントリを示している。4 つのキャッシュエントリ右端の 1 又は 0 は、それぞれ使用フラグの値である。この 4 つの使用フラグ U を  $U_0 \sim U_3$  と記す。

#### 【0 0 6 4】

同図上段では  $(U_0 \sim U_3) = (1, 0, 1, 0)$  であるので、ウェイ 0、2 のキャッシュエントリはアクセスがあったことを、ウェイ 1、3 のキャッシュエントリはアクセスがないことを意味する。

この状態で、メモリアクセスがセット N 内のウェイ 1 のキャッシュエントリにヒットした場合、同図中段に示すように、 $(U_0 \sim U_3) = (1, 1, 1, 0)$  に更新される。つまり、実線に示すようにウェイ 1 の使用フラグ  $U_1$  が 0 から 1 に更新される。

#### 【0 0 6 5】

さらに、同図中段の状態で、メモリアクセスがセット N 内のウェイ 3 のキャッシュエントリにヒットした場合、同図下断に示すように、 $(U_0 \sim U_3) = (0, 0, 0, 1)$  に更新される。つまり、実線に示すようにウェイ 3 の使用フラグ  $U_3$  が 0 から 1 に更新される。加えて、破線に示すようにウェイ 3 以外の使用フラグ  $U_0 \sim U_2$  が 1 から 0 に更新される。これにより、ウェイ 3 のキャッシュエントリが、ウェイ 0 ～ 2 の各キャッシュエントリよりも新しくアクセスされたことを意味することになる。

#### 【0 0 6 6】

リプレース部 1 4 0 は、キャッシュミス時に  $W=1$  のキャッシュエントリが存在しなければ、使用フラグに基づいてリプレース対象のキャッシュエントリを決定してリプレースを行う。例えば、フラグ更新部 3 9 は、図 5 上段では、ウェイ 1 とウェイ 3 の何れかをリプレース対象と決定し、図 5 中段ではウェイ 3 をリプレース対象と決定し、図 5 下段ではウェイ 0 ～ 2 の何れかをリプレース対象と決定する。

#### 【0 0 6 7】

<ウィークフラグ W の説明>

図 1 3 (a) ウィークフラグが存在しないと仮定した場合の比較例であり、キャッシュエントリがリプレースされる様子を示す図である。同図においても、図 1 2 と同様にウェイ 0 ～ 3 に跨るセット N を構成する 4 つのキャッシュエントリを示している。、4 つのキャッシュエントリ右端の 1 又は 0 は、それぞれ使用フラグの値である。また、データ E のみアクセス頻度の低いデータを、データ A、B、C、D はアクセス頻度の高いデータとする。

#### 【0 0 6 8】

同図 (a) の第 1 段目の状態で、プロセッサ 1 がデータ E にアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、 $U=0$  のキャッシュエントリの中からアクセス頻度の高いデータ C のキャッシュエントリがアクセス頻度の低いデー

タ E にリプレースされ、第 2 段目の状態となる。

第 2 段目の状態で、プロセッサ 1 がデータ C にアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、 $U=0$  のキャッシュエントリーであるアクセス頻度の高いデータ D のキャッシュエントリーがアクセス頻度の高いデータ C にリプレースされ、第 3 段目の状態となる。

【0 0 6 9】

第 3 段目の状態で、プロセッサ 1 がデータ D にアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、アクセス頻度の高いデータ C のキャッシュエントリーがアクセス頻度の高いデータ D にリプレースされ、第 3 段目の状態となる。

同様に、第 4 段目でも、使用頻度の低いデータ E はリプレース対象として選択されないで、キャッシュメモリーに残っている。

【0 0 7 0】

第 5 段目の状態で、使用頻度の低いデータ E は最も古い ( $U=0$ ) ことから、リプレース対象として選択されて、追い出される。

このように、擬似 L R U 方式において (通常の L R U 方式においても)、アクセス頻度の低いデータ E によって、4 ウェイの場合は最悪 4 回のキャッシュミスを誘発する可能性がある。

【0 0 7 1】

図 1 3 (b) は、リプレース処理におけるウィークフラグ W の役割を示す説明図である。

同図 (b) の第 1 段目の状態 (同図 (a) の第 1 段目と同じ) で、プロセッサ 1 がデータ E にアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、 $U=0$  のキャッシュエントリーの中からアクセス頻度の高いデータ C のキャッシュエントリーがアクセス頻度の低いデータ E にリプレースされる。このとき、プロセッサ 1 は、データ E のキャッシュエントリーにウィークフラグ W を 1 に設定するものとする。これにより、次のキャッシュミス時にデータ E のキャッシュエントリーが真っ先に追い出され、第 2 段目の状態となる。

【0 0 7 2】

第 2 段目の状態で、プロセッサ 1 がデータ C にアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、 $W=1$  のキャッシュエントリーであるアクセス頻度の低いデータ E のキャッシュエントリーがリプレース対象として選択され、アクセス頻度の高いデータ C にリプレースされ、第 3 段目の状態となる。

このように、ウィークフラグ W を設けることにより、アクセス頻度の低いデータによるキャッシュミスの誘発を低減することができる。

【0 0 7 3】

< U フラグ更新処理 >

図 1 4 は、リプレース部 1 4 0 における U フラグ更新処理を示すフローチャートである。

同図では、バリッドフラグが 0 (無効) であるキャッシュエントリーの使用フラグ U は 0 に初期化されているものとする。

【0 0 7 4】

同図において、リプレース部 1 4 0 は、キャッシュヒットしたとき (ステップ S 6 1)、セットインデックスにより選択されたセットにおけるヒットしたウェイの使用フラグ U を 1 にセットし (ステップ S 6 2)、そのセット内の他のウェイの使用フラグ U を読み出し (ステップ S 6 3)、読み出した使用フラグ U が全て 1 であるか否かを判定し (ステップ S 6 4)、全て 1 でなければ終了し、全て 1 であれば他のウェイの全ての使用フラグ U を 0 にリセットする (ステップ S 6 5)。

【0 0 7 5】

このようにしてリプレース部 1 4 0 は、図 1 2、図 1 3 (a) (b) に示した更新例のように、使用フラグ U を更新する。



## 【0076】

## ＜リプレース処理＞

図15は、リプレース部140におけるリプレース処理を示すフローチャートである。同図においてリプレース部140は、メモリアクセスがミスしたとき（ステップS91）、セットインデックスにより選択されたセットにおける、4つウェイの使用フラグU及びウィークフラグWを読み出し（ステップS92）、W=1のウェイが存在するか否かを判定する（ステップS93）。W=1のウェイが存在しないと判定された場合、U=0のウェイを1つ選択する（ステップS94）。このとき、使用フラグUが0になっているウェイが複数存在する場合は、リプレース部140はランダムに1つを選択する。また、W=1のウェイが存在すると判定された場合、Uフラグの値に関わらずW=1のウェイを1つ選択する（ステップS95）。このとき、ウィークフラグWが1になっているウェイが複数存在する場合は、リプレース部140はランダムに1つを選択する。

## 【0077】

さらに、リプレース部140は、当該セットにおける選択されたウェイのキャッシュエントリを対象にリプレースし（ステップS96）、リプレース後に当該キャッシュエントリの使用フラグUを1に、ウィークフラグWを0初期化する（ステップS97）。なお、このときバリッドフラグV、ダーティフラグDは、それぞれ1、0に初期化される。

このように、W=1のウェイが存在しない場合、リプレース対象は、使用フラグUが0のキャッシュエントリの中から1つ選択される。

## 【0078】

また、W=1のウェイが存在する場合、リプレース対象は、使用フラグUが0であると1であることを問わず、W=1のウェイのキャッシュエントリから1つ選択される。これにより図13（a）（b）に示したように、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。

## 【0079】

## ＜フラグ改変部141の構成＞

図16は、フラグ改変部141の構成を示すブロック図である。同図の構成は、図5に示したフラグ改変部41と比較して、フラグ書換部407の代わりにフラグ書換部407aを備える点が異なっている。

## 【0080】

フラグ書換部407aは、フラグ書換部407の機能に加えて、Wフラグの改変と使用フラグUの改変とを行う点が異なっている。そのため、コマンドレジスタ401には、Wフラグの設定を指示するWコマンド、Uフラグの設定を指示するUコマンドがプロセッサ1によって設定可能になっている。図17に、これらのコマンドフォーマットの一例を示す。同図のコマンドフォーマットは、図6（d）に示したコマンドフォーマットに対してWコマンド及びUコマンドのフィールドが追加されている。Wコマンド及びUコマンドの内容はDコマンドやVコマンドと同様である。

## 【0081】

## ＜Wフラグ設定処理＞

図18は、フラグ書換部407aにおけるWフラグ設定処理の一例を示すフローチャートである。

フラグ書換部407aは、コマンドレジスタ401にWフラグ設定コマンドが保持されている場合、スタートラインからエンドラインまでの各ラインアドレスを順に出力しながらループ1の処理（S82～S86）を行う。フラグ書換部407aは、各ラインについて同じ処理を行うので、ここでは1ライン分の処理について説明する。

## 【0082】

すなわち、フラグ書換部407aは、キャッシュメモリ3がプロセッサ1からアクセスされていない間に、ラインアドレスをアドレスレジスタ20に出力し（S83）、アドレスレジスタ20のタグアドレスとキャッシュエントリのタグとを比較器32a～32dに比較させ、ヒットするかどうかを判定する（S84）。さらにフラグ書換部407aは

、ヒットした場合には、ヒットしたキャッシュエントリーに対してWフラグを1にセットし（S85）、ミスヒットした場合には、キャッシュメモリにエントリーされていないのでなにもしない。

これにより、スタートラインからエンドラインまでの各ラインについて、キャッシュメモリ3にエントリーされている場合には、Wフラグが1に設定される。

#### 【0083】

##### <Uフラグ設定処理>

フラグ書換部407aは、コマンドレジスタ401にUフラグ設定コマンドが保持されている場合、当該コマンドに従ってUフラグを設定する。この処理は、図18においてWフラグをUフラグと読み替えることにより、Wフラグ設定処理と全く同様に実行される。

#### 【0084】

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、Wフラグを設定することにより、W=1のキャッシュエントリーをキャッシュミス時には最古のキャッシュエントリとして真っ先にリプレース対象とすることができる。また、使用フラグUの値は1ビットだけであるがアクセス順序が古いか新しいかをしめすので、Uフラグ設定処理によって、スタートラインからエンドラインまでの各ラインについてアクセス順序を設定することができる。例えば、プロセッサ1は、キャッシュメモリ3に残しておきたいアドレスのデータをアドレス範囲として指定してU=1を設定するUフラグ設定コマンドを発行し、

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、Wフラグを設定することにより、W=1のキャッシュエントリーをキャッシュミス時には最古のキャッシュエントリとして真っ先にリプレース対象とすることができる。また、使用フラグUの値は1ビットだけであるがアクセス順序が古いか新しいかをしめすので、Uフラグ設定処理によって、スタートラインからエンドラインまでの各ラインについてアクセス順序を設定することができる。例えば、プロセッサ1は、キャッシュメモリ3に残しておきたいアドレスのデータをアドレス範囲として指定してU=1を設定するUフラグ設定コマンドを発行すればよい。逆に、キャッシュメモリ3か追い出してもよいデータをアドレス範囲として指定してU=0を設定するUフラグ設定コマンドを発行すればよい。

#### 【0085】

##### <変形例>

(1) W=1のキャッシュエントリーは真っ先にリプレース対象とされるが、制御部は、リプレースされるまでの間に、ダーティであればクリーニング（ライトバック）を行ってもよい。

(2) 図6（a）（b）（c）に示した各命令は、コンパイラによりプログラム中に挿入してもよい。その際、コンパイラは、例えば配列データの書き込みや、圧縮動画データをデコードする際のブロックデータの書き込み等、これ以上書き込みをしないプログラム位置に、上記各命令を挿入するようにすればよい。

##### 【産業上の利用可能性】

#### 【0086】

本発明は、メモリアクセスを高速化するためのキャッシュメモリに適しており、例えば、オンチップキャッシュメモリ、オフチップキャッシュメモリ、データキャッシュメモリ、命令キャッシュメモリ等に適している。

##### 【図面の簡単な説明】

#### 【0087】

【図1】本発明の実施の形態1におけるプロセッサ、キャッシュメモリ、メモリを含むシステムの概略構成を示すブロック図である。

【図2】キャッシュメモリの構成例を示すブロック図である。

【図3】キャッシュエントリーの詳細なビット構成を示す図である。

【図4】制御部の構成を示すブロック図である。

【図5】フラグ改変部の構成例を示すブロック図である。



【図 6】 (a) スタートアドレスレジスタにスタートアドレスを書き込む命令の一例を示す。(b) サイズレジスタにサイズを書き込む命令の一例を示す。(c) コマンドレジスタにコマンドを書き込む命令の一例を示す。(d) コマンドの一例を示す。

【図 7】 スタートアライナ及びエンドアライナの説明図である。

【図 8】 フラグ書換部 4 0 7 におけるフラグ改変処理の一例を示すフローチャートである。

【図 9】 本発明の実施の形態 2 におけるキャッシュメモリの構成を示すブロック図である。

【図 1 0】 キャッシュエントリーのビット構成を示す。

【図 1 1】 制御部の構成を示すブロック図である。

【図 1 2】 フラグ更新部による使用フラグ U の更新例を示す。

【図 1 3】 (a) ウィークフラグが存在しない場合にキャッシュエントリーがリプレイスされる様子を示す図である。(b) リプレイス処理におけるウィークフラグ W の役割を示す説明図である。

【図 1 4】 リプレイス部における U フラグ更新処理を示すフローチャートである。

【図 1 5】 リプレイス部におけるリプレイス処理を示すフローチャートである。

【図 1 6】 フラグ改変部の構成を示すブロック図である。

【図 1 7】 コマンドフォーマットの一例を示す。

【図 1 8】 フラグ書換部における W フラグ設定処理の一例を示すフローチャートである。

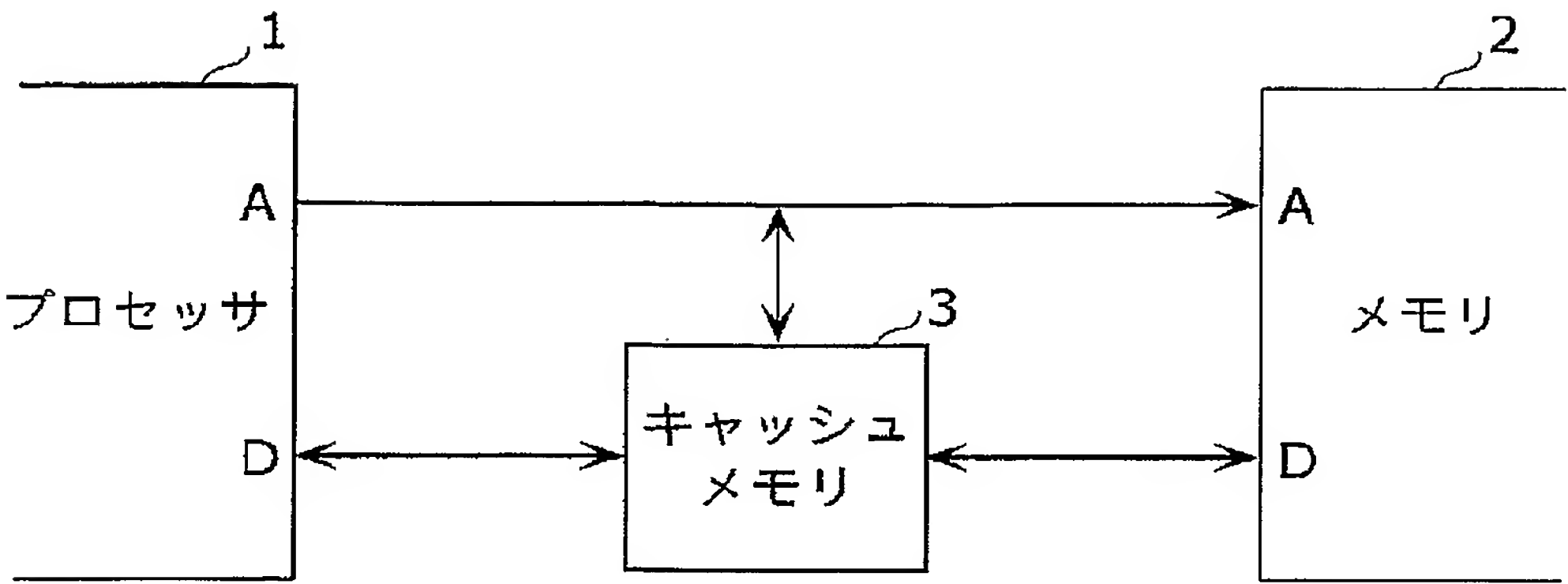
#### 【符号の説明】

##### 【0 0 8 8】

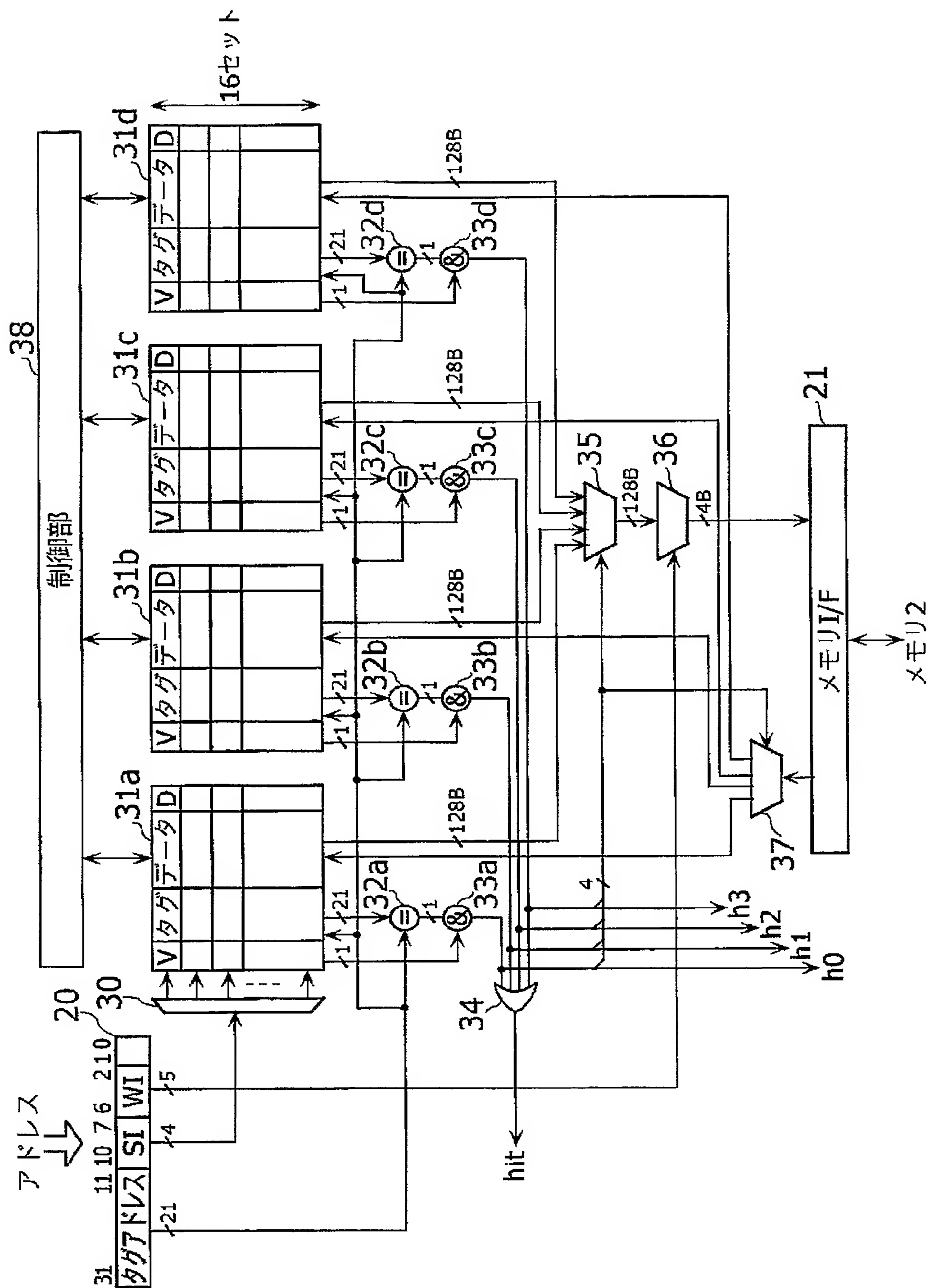
1	プロセッサ
2	メモリ
3	キャッシュメモリ
2 0	アドレスレジスタ
2 1	メモリ I / F
3 0	デコーダ
3 1 a ~ 3 1 d	ウェイ
3 2 a ~ 3 2 d	比較器
3 3 a ~ 3 3 d	アンド回路
3 4	オア回路
3 5	セレクタ
3 6	セレクタ
3 7	デマルチプレクサ
3 8	制御部
3 9	フラグ更新部
4 0	リプレイス部
4 1	フラグ改変部
1 3 1 a ~ 1 3 1 d	ウェイ
1 3 8	制御部
1 3 9	フラグ更新部
1 4 0	リプレイス部
1 4 1	フラグ改変部
4 0 1	コマンドレジスタ
4 0 2	スタートアドレスレジスタ
4 0 3	サイズレジスタ
4 0 4	加算器
4 0 5 a	スタートアライナ
4 0 5 b	スタートアライナ

4 0 6 a エンドアライナ  
4 0 6 b エンドアライナ  
4 0 7 フラグ書換部  
4 0 7 a フラグ書換部

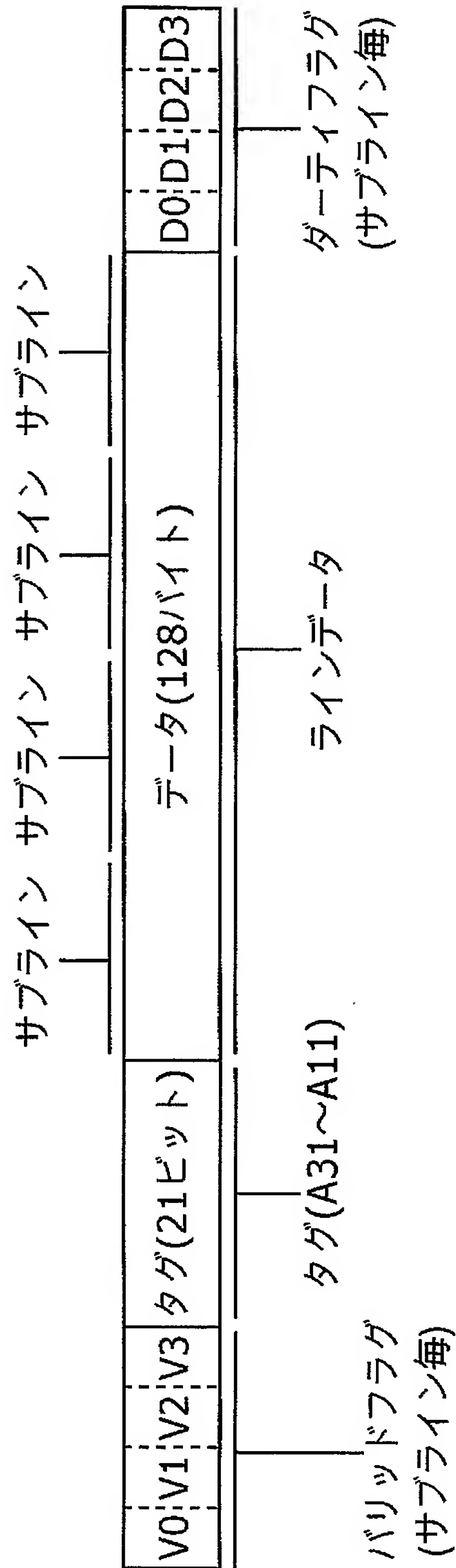
【書類名】 図面  
【図 1】



【図 2】

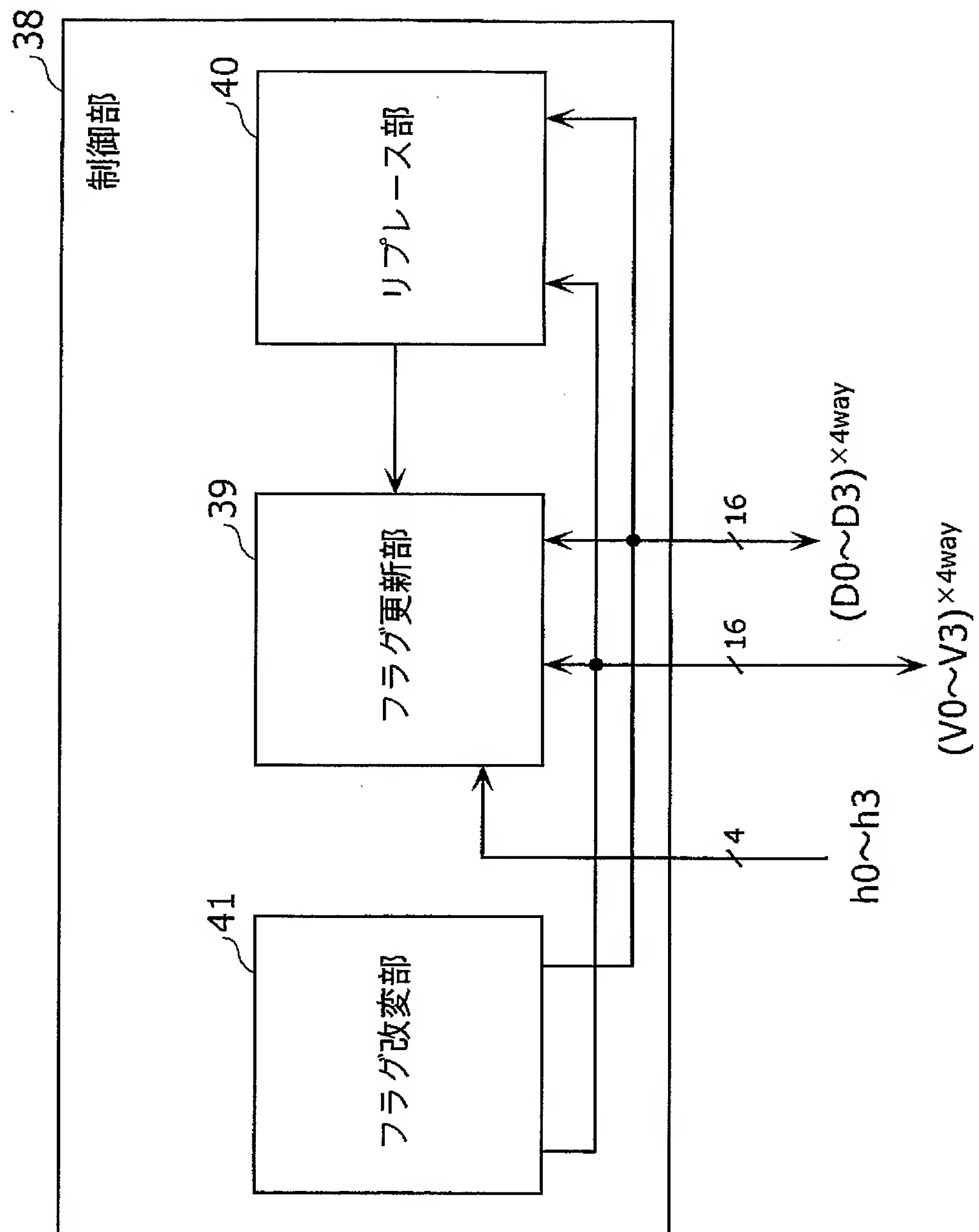


【図 3】

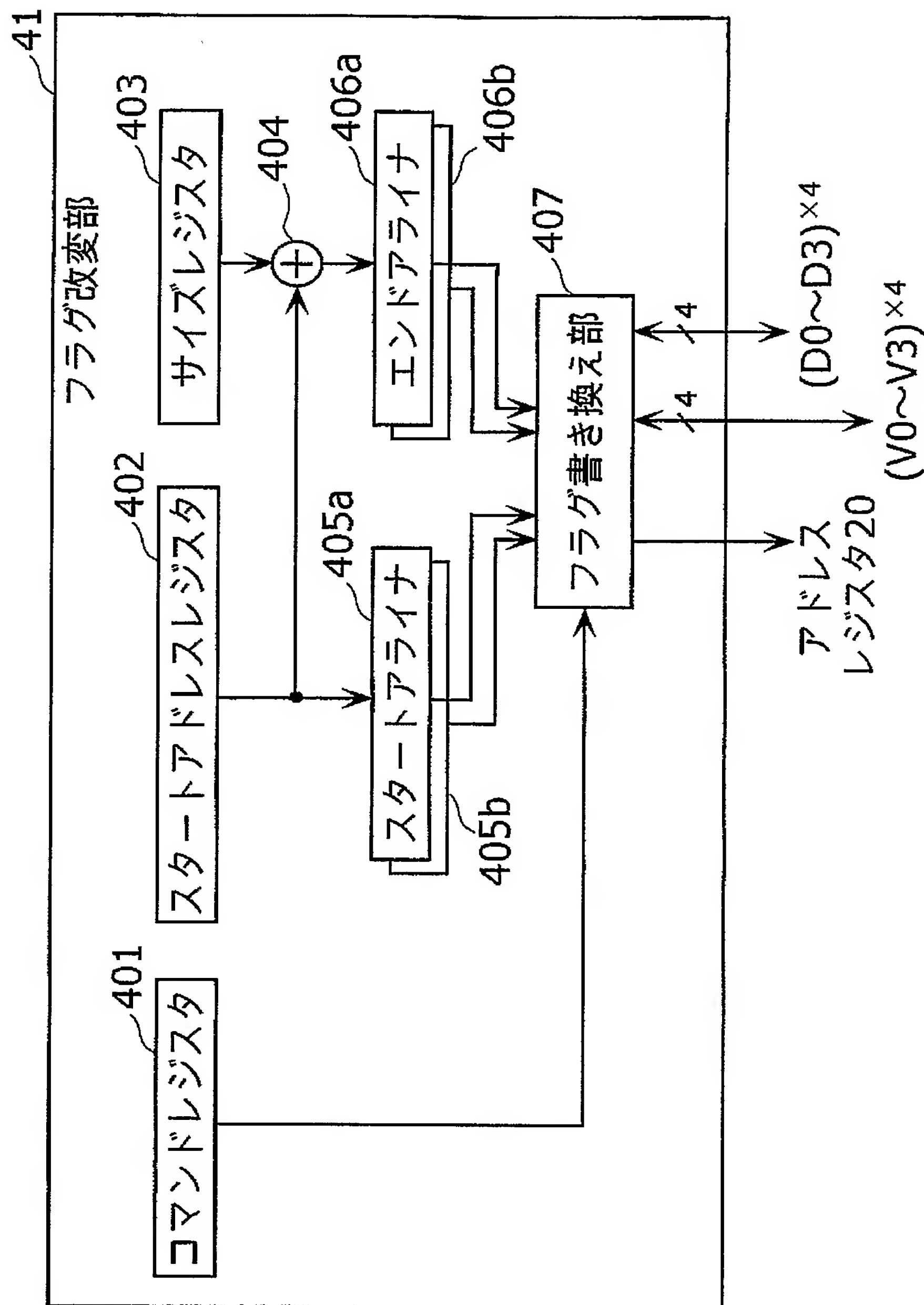




【図 4】



【図 5】



(a)

(デスティネーション)スタートアドレスレジスタを指定  
 (ソース)スタートアドレスを指定  
 転送命令

(b)

転送命令  
 (デスティネーション)サイズレジスタを指定  
 (ソース)サイズを指定

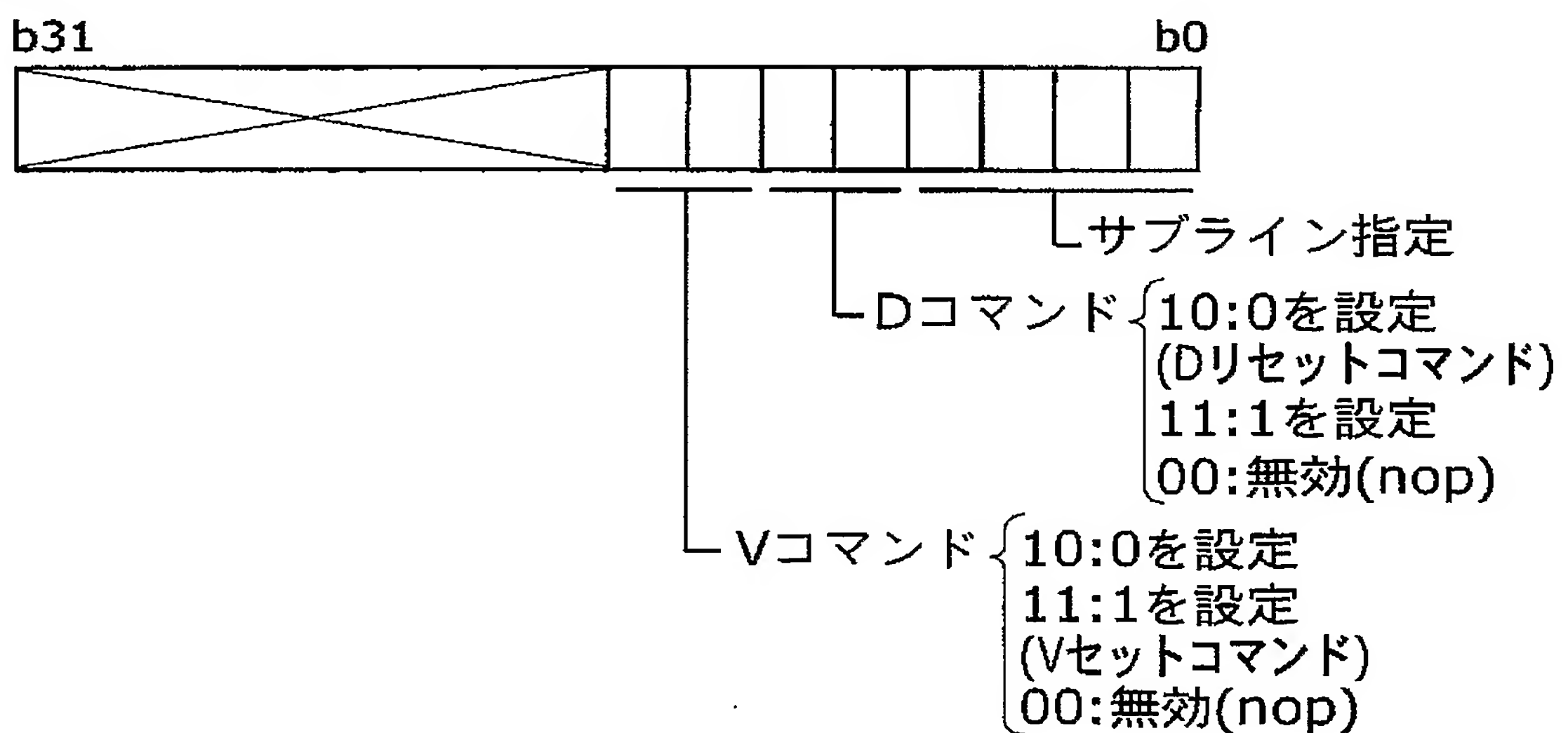
(C)

↑ (ソース) コマンド を指定

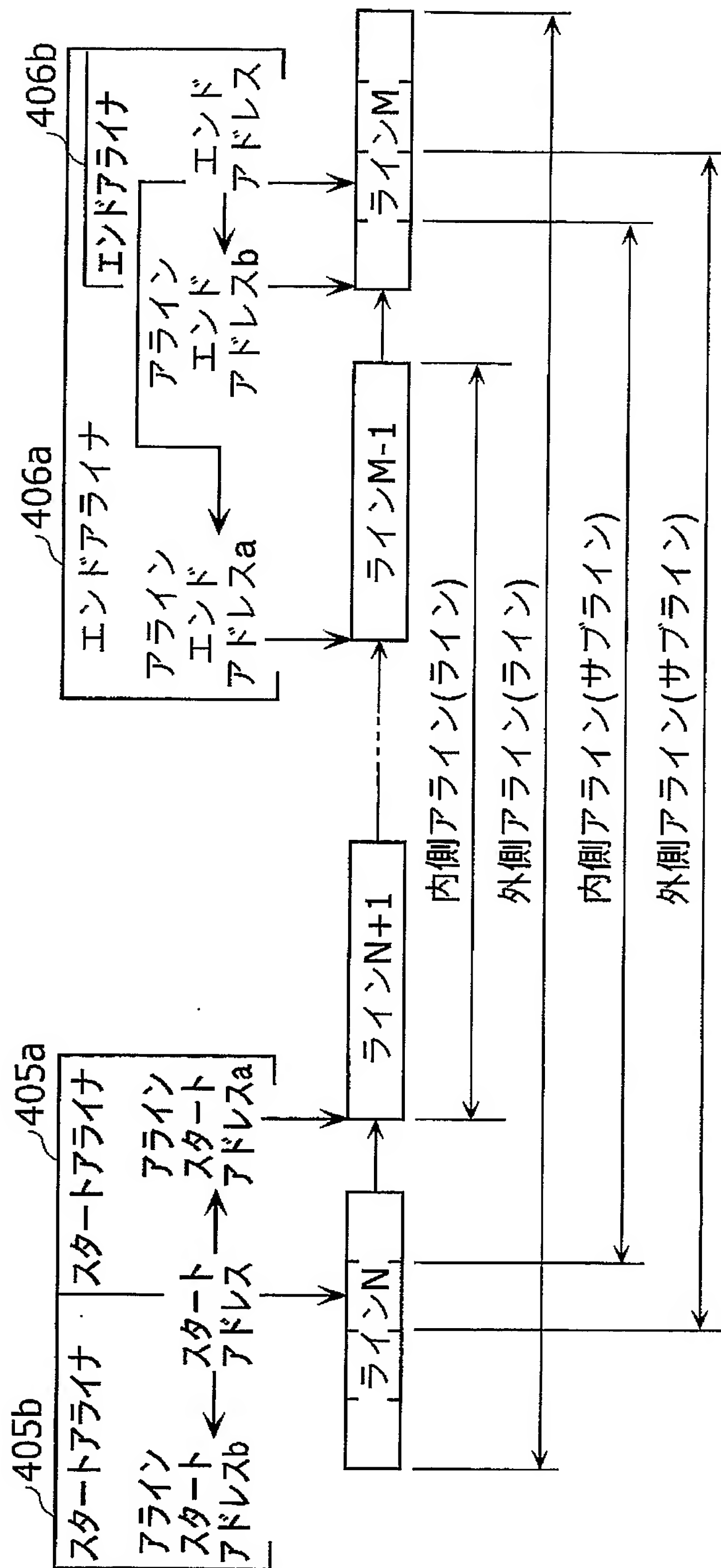
↑ (デスティネーション) コマンドレジスタを指定

↑ 転送命令

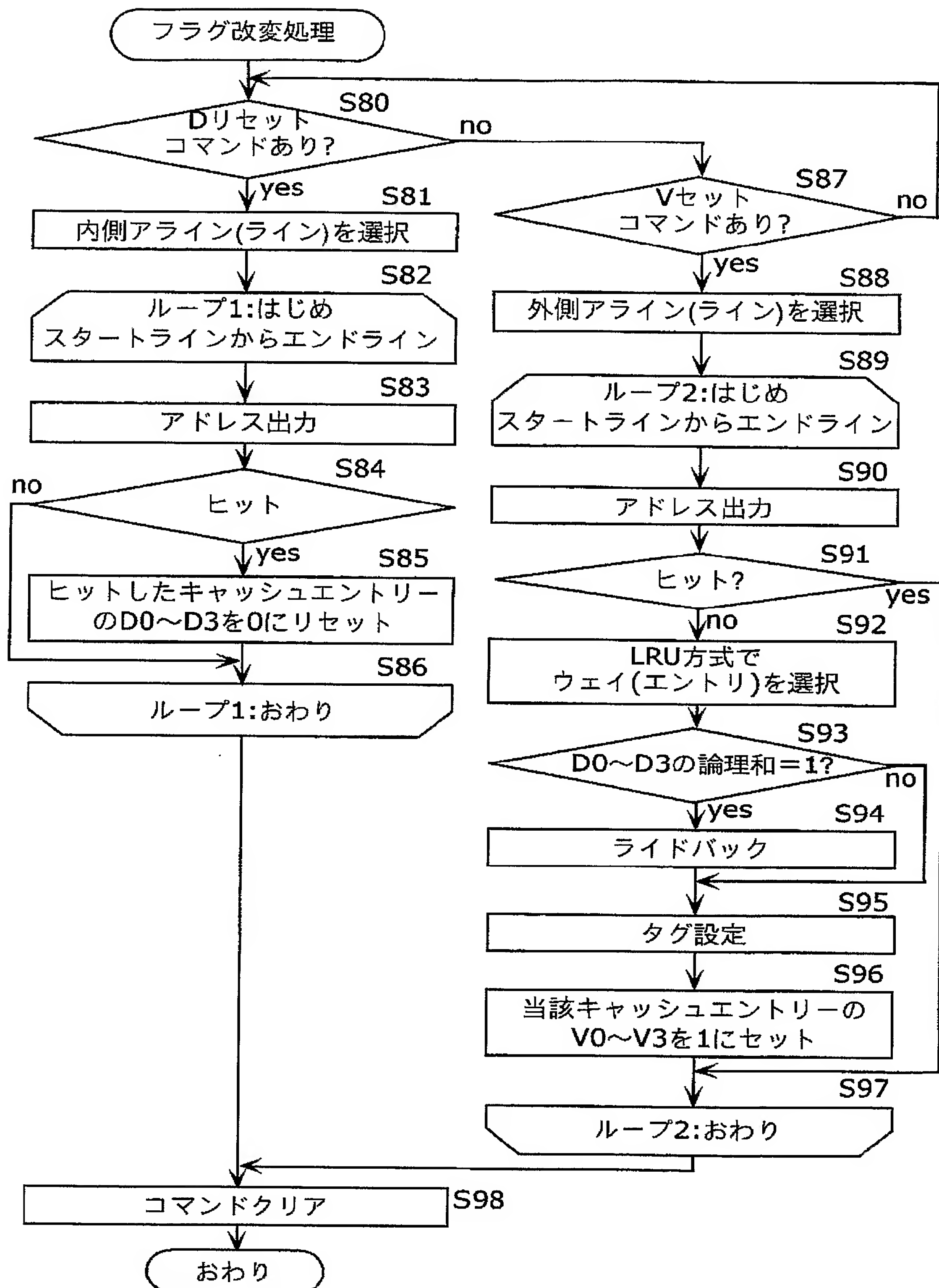
(d)



【図 7】

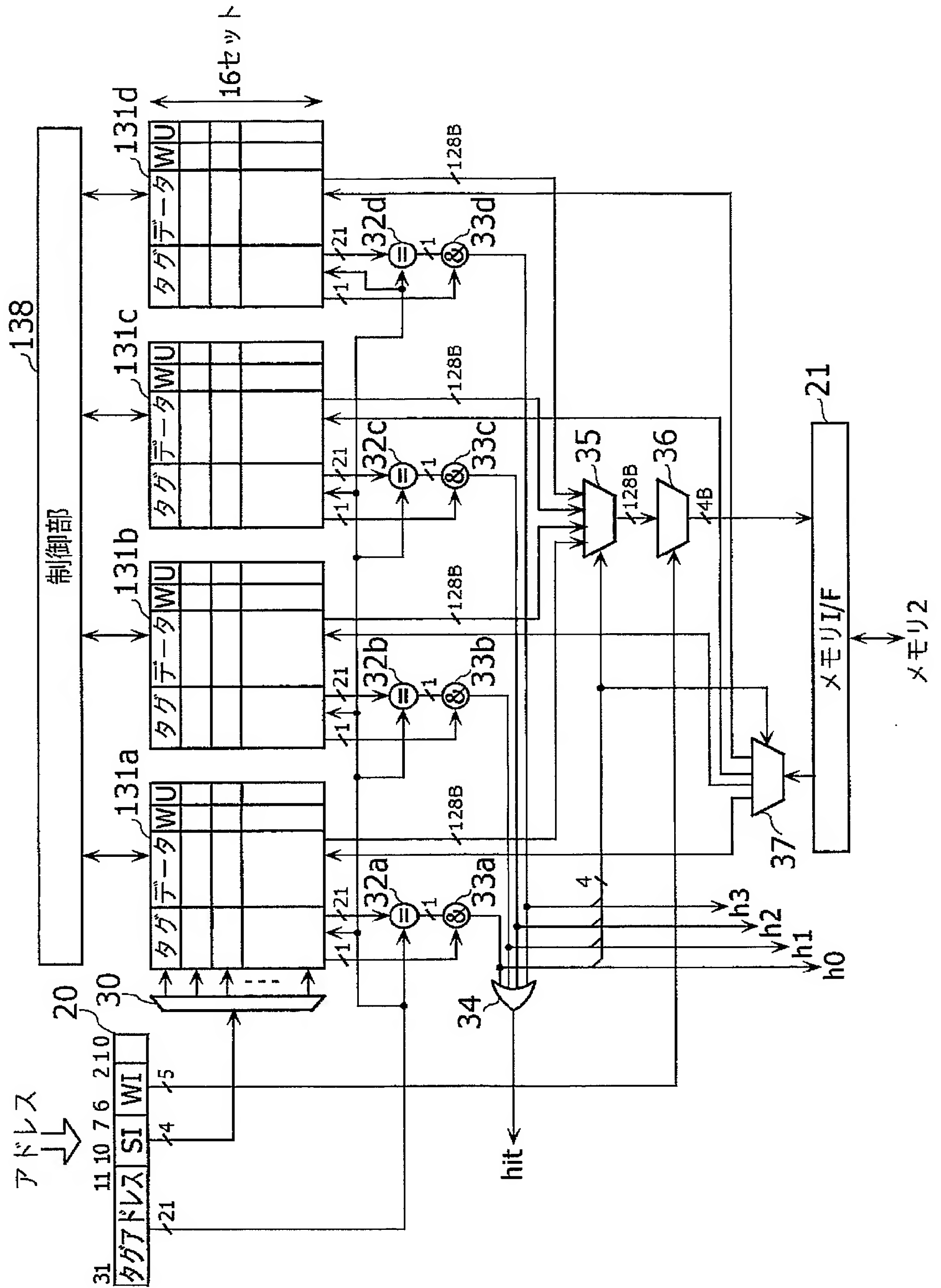


【図 8】

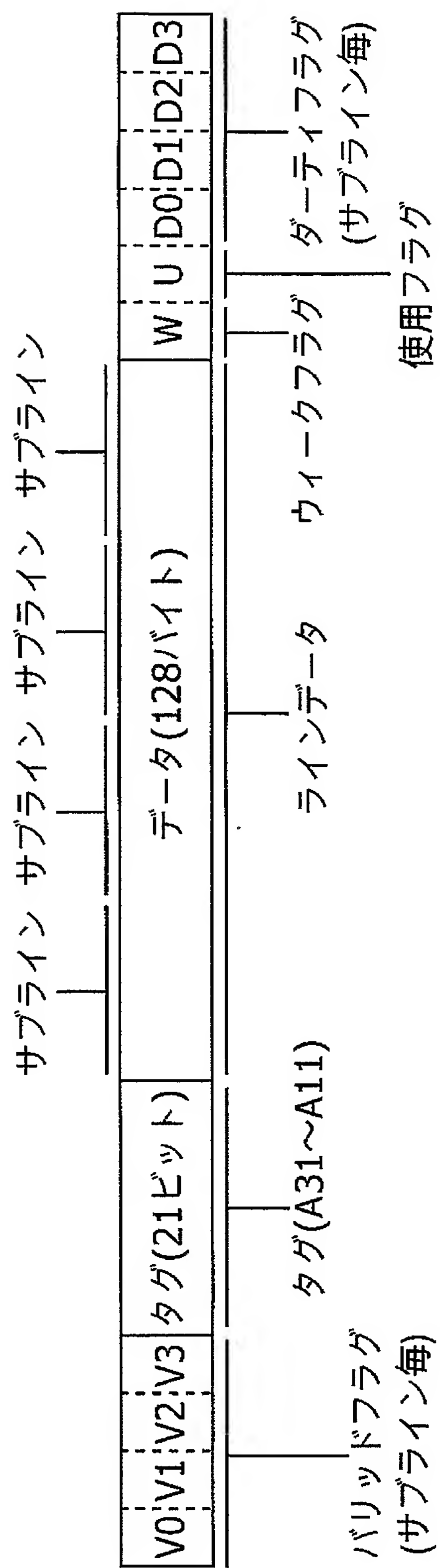




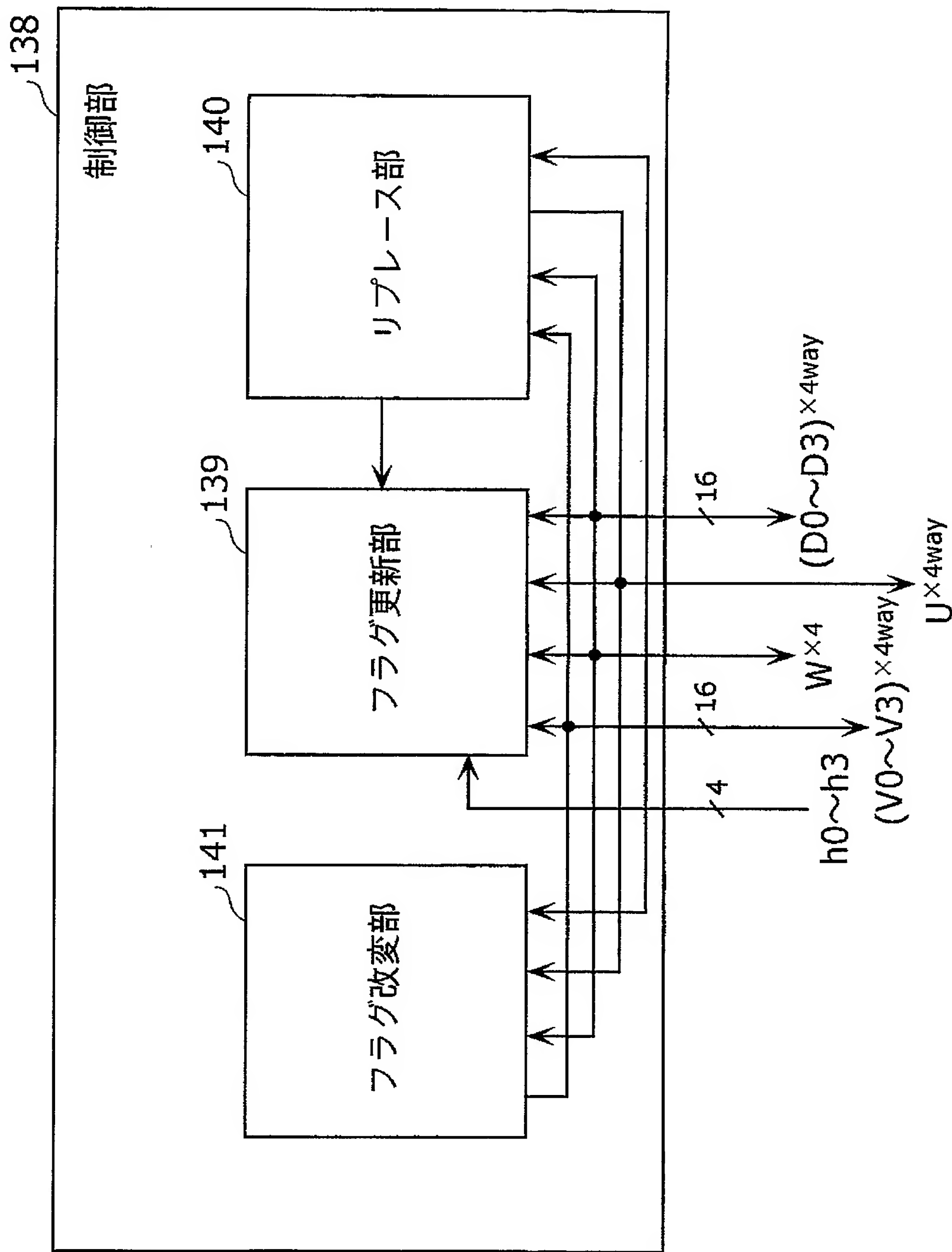
【図 9】



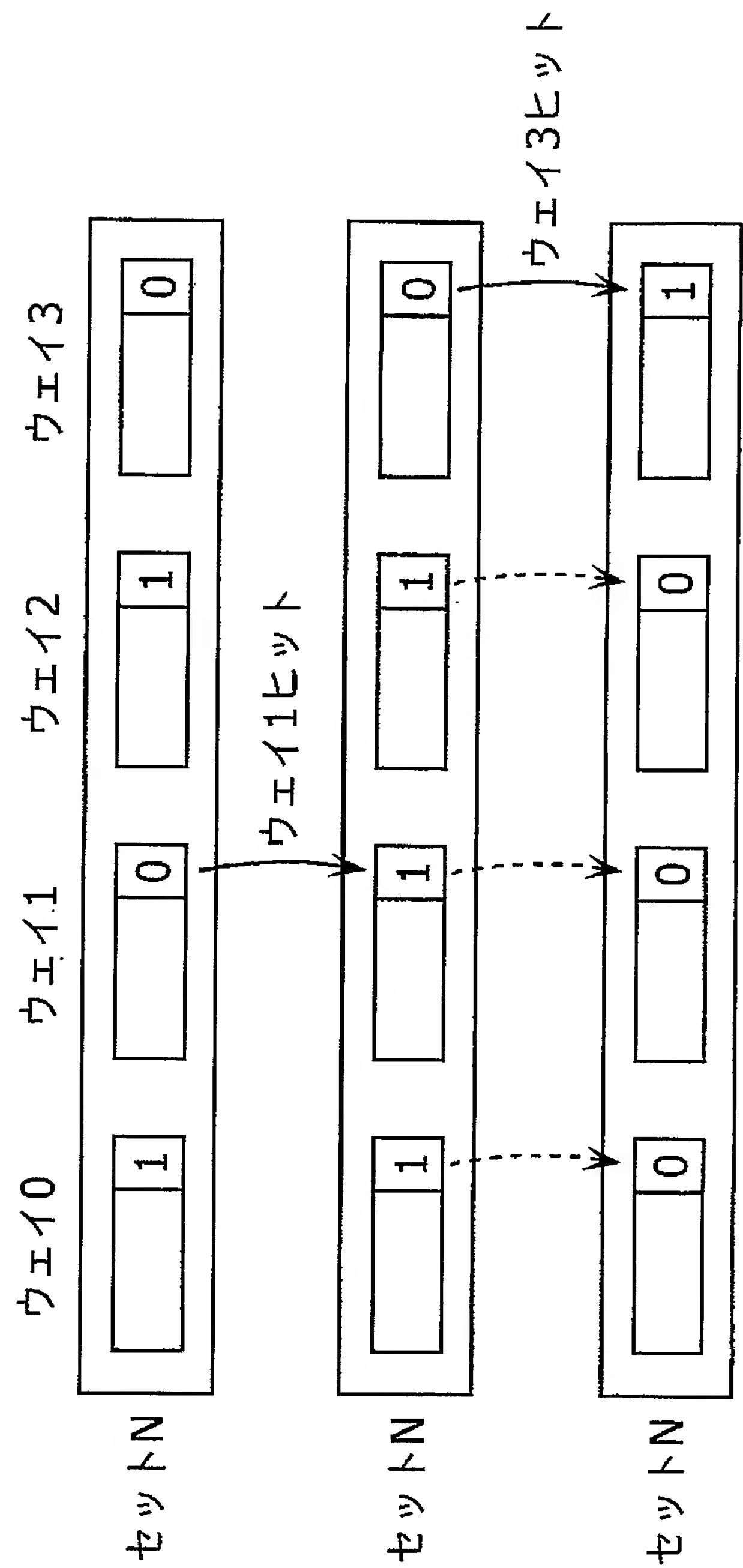
【図 1 0】



【図 1 1】

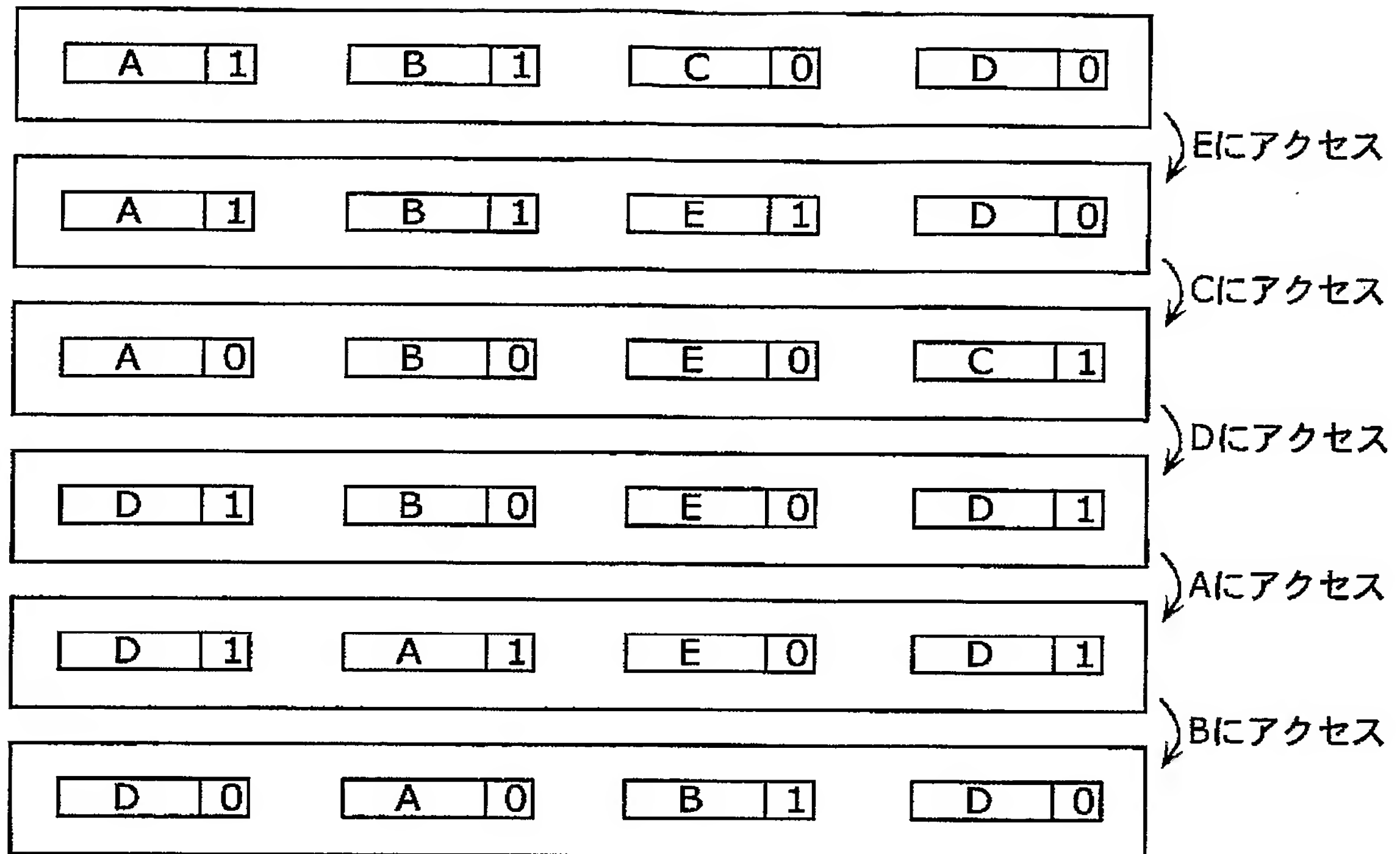


【図 1 2】

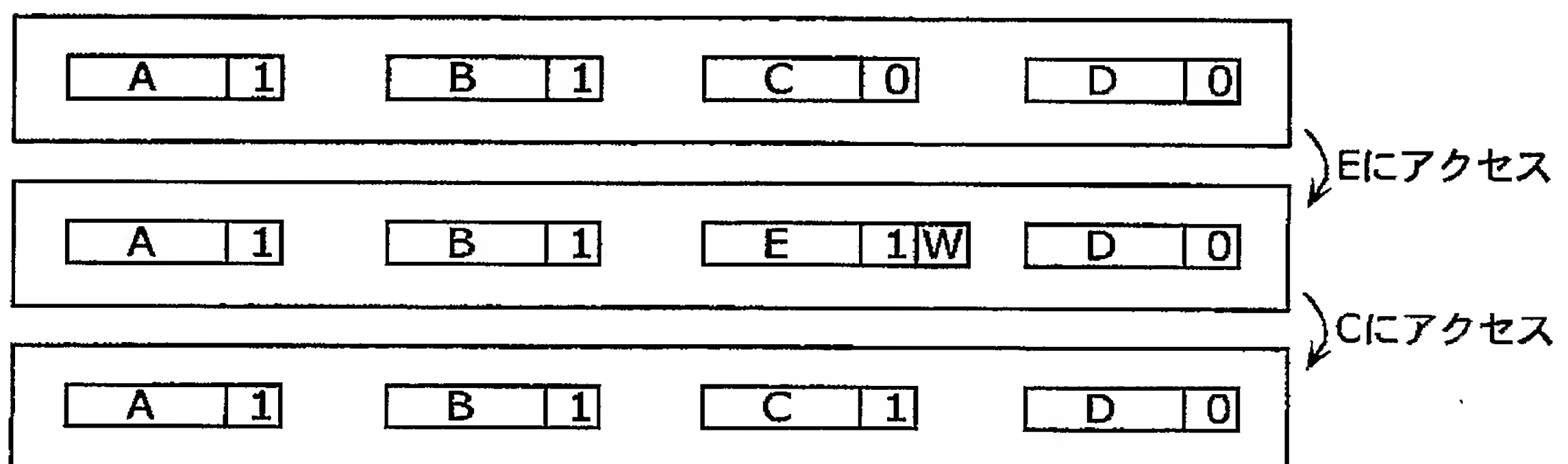


【図 1 3】

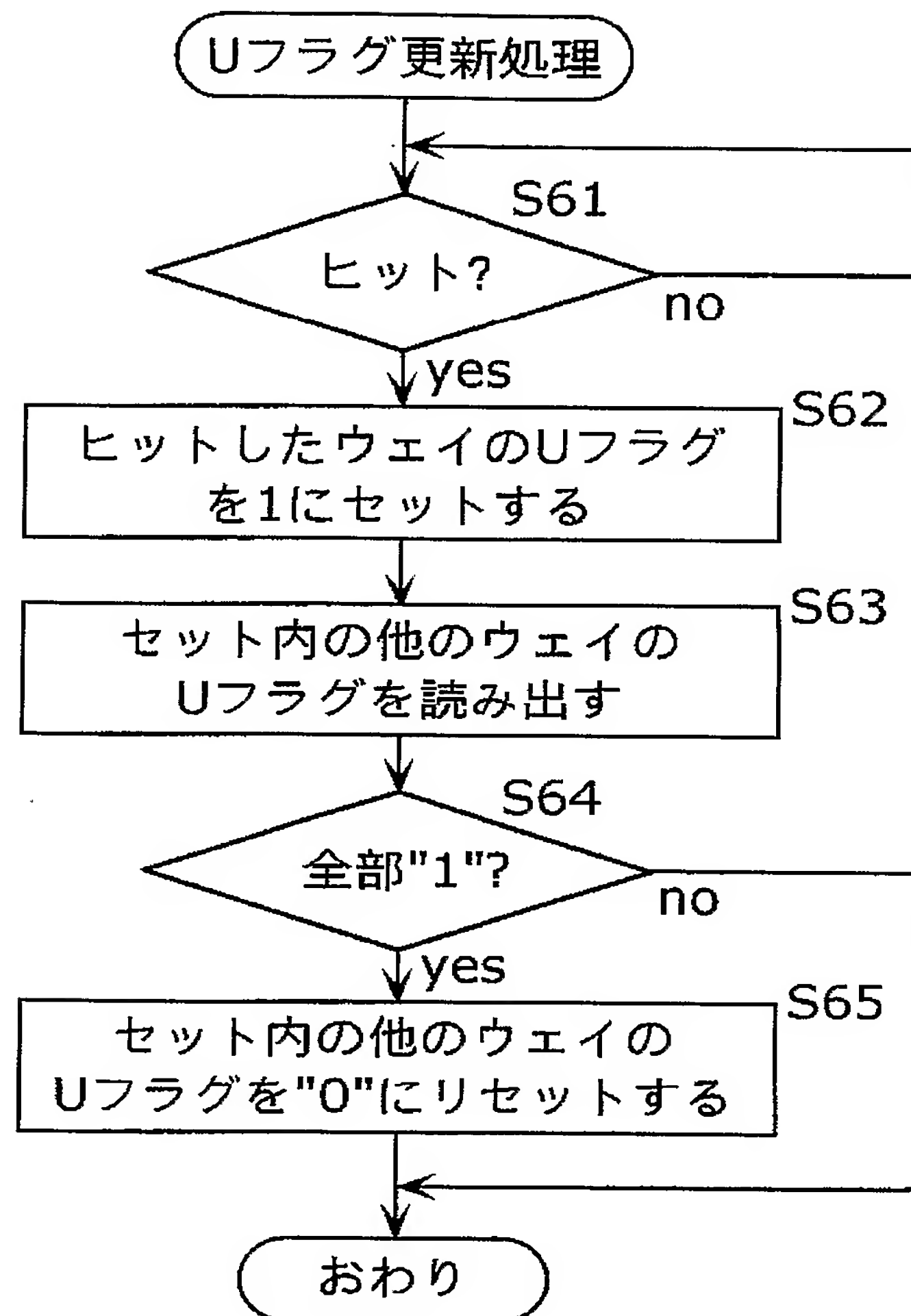
(a)



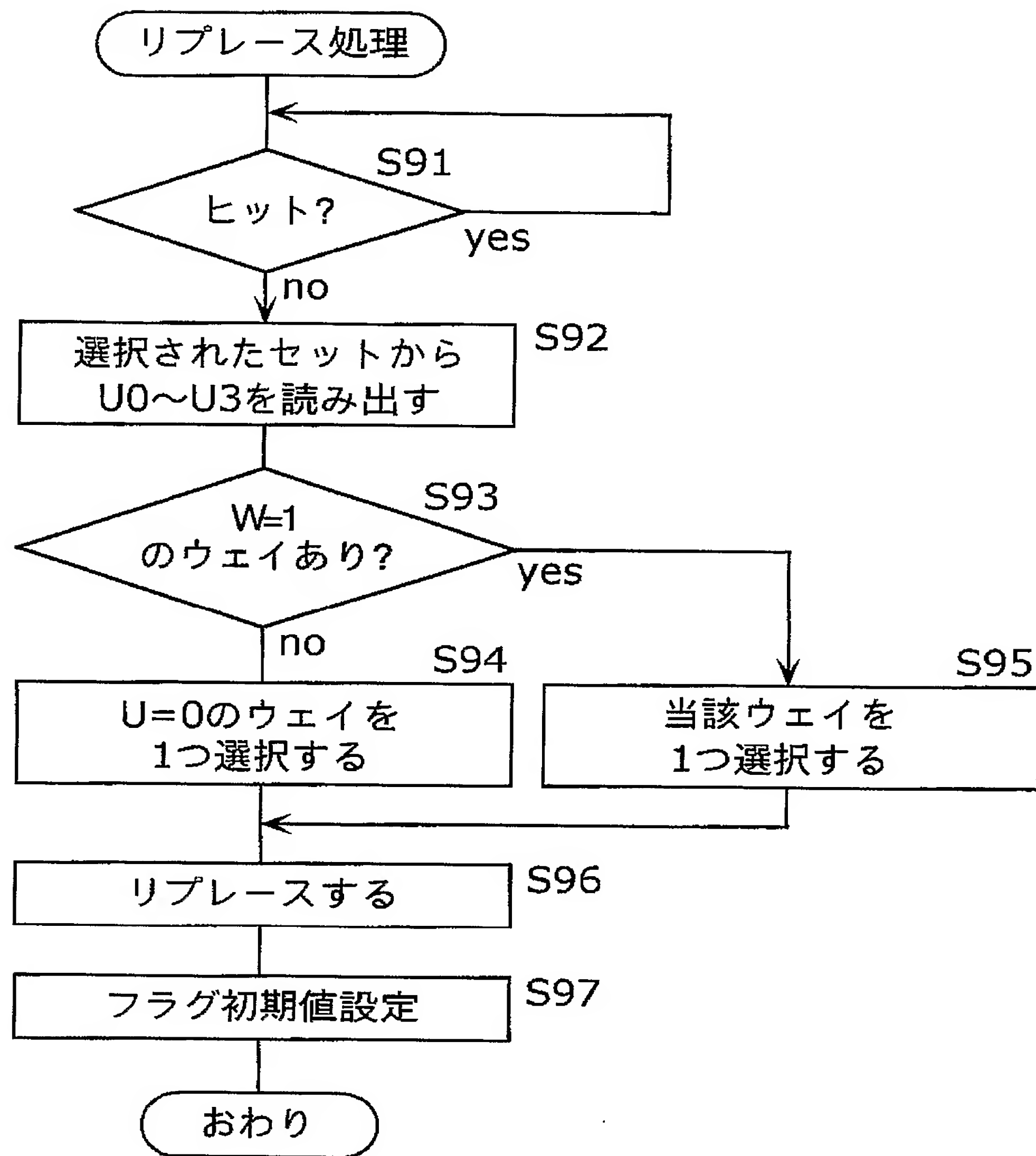
(b)



【図 1 4】

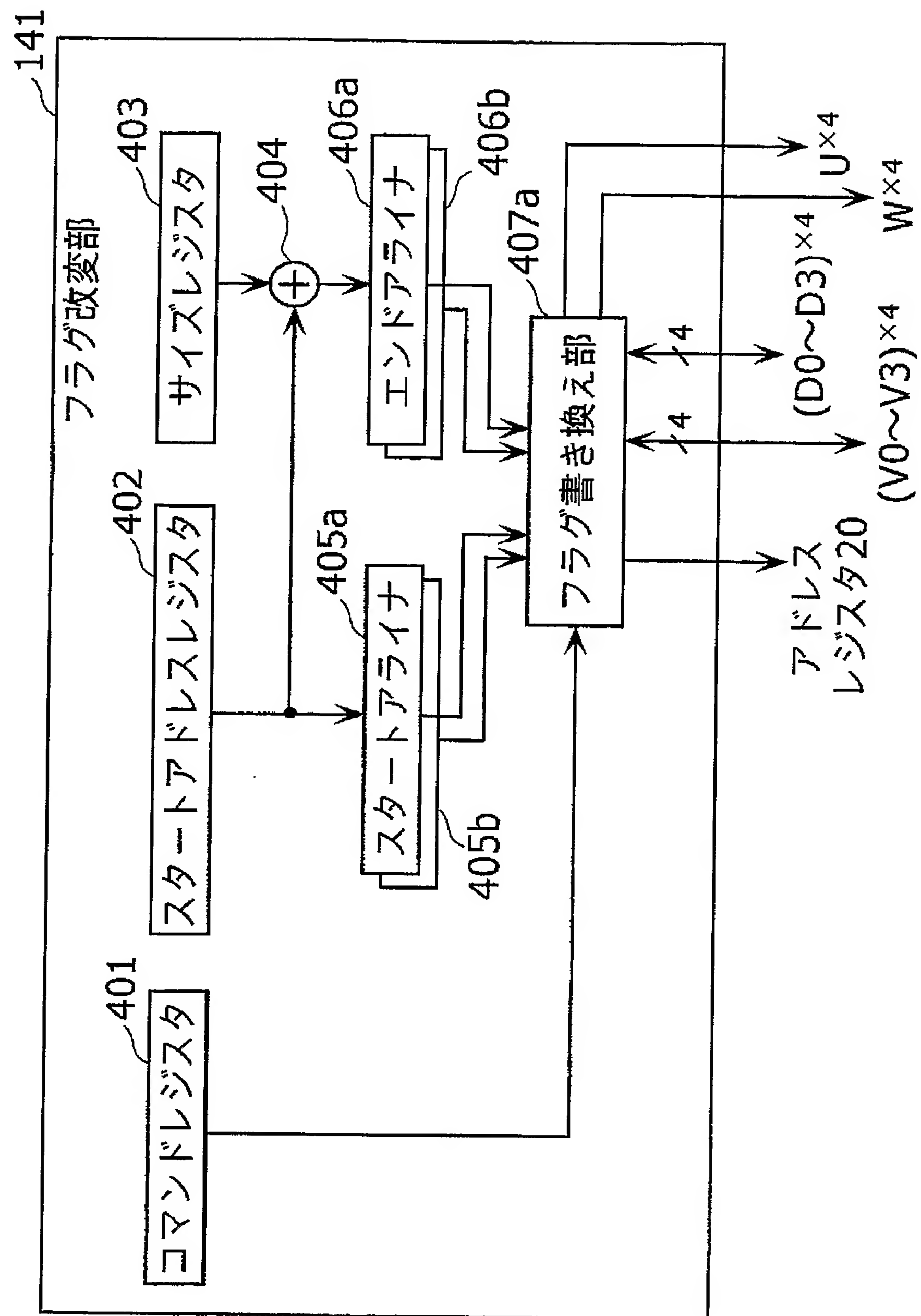


【図 15】

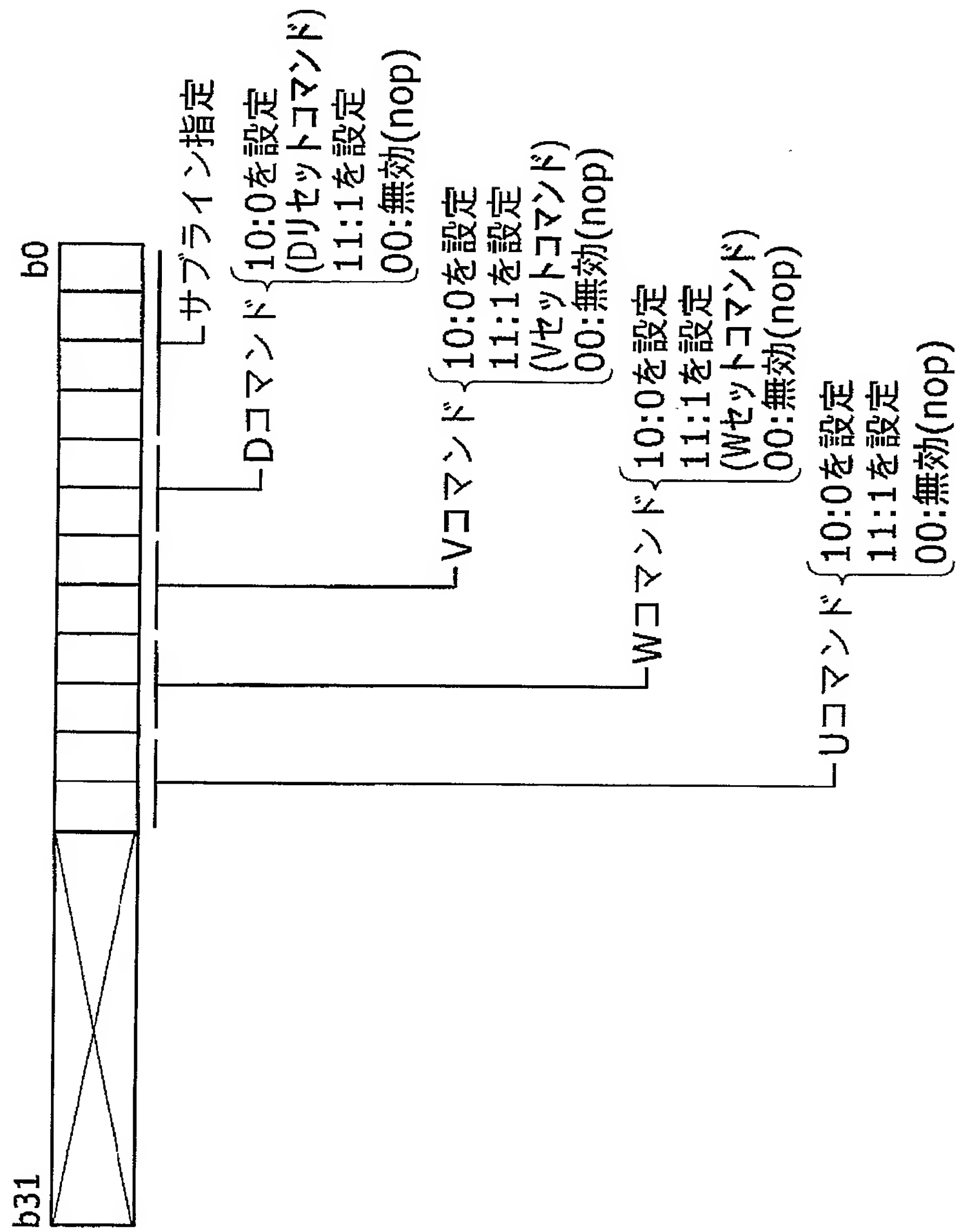




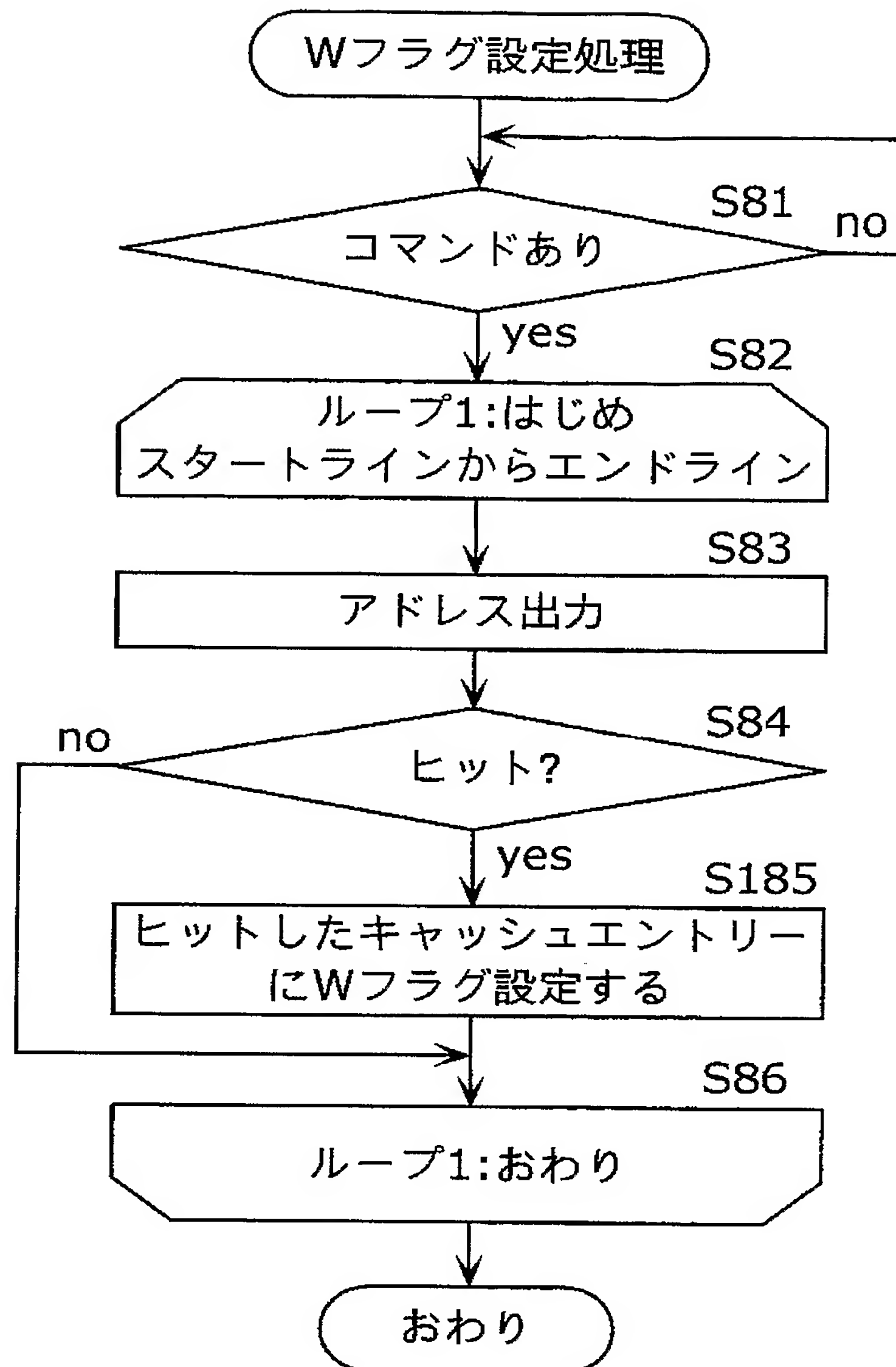
【図 16】



【図 1 7】



【図 18】



【書類名】 要約書

【要約】

【課題】 無駄なリプレイスや無駄なライトバックを防止するキャッシュメモリを提供する。

【解決手段】 キャッシュの単位データを保持するキャッシュエントリーに対応させて、当該キャッシュエントリーが有効であるか否かを示すバリッドフラグと、当該キャッシュエントリーに対する書き込みがなされたか否かを示すダーティフラグと有するキャッシュメモリであって、プロセッサからの指示に基づき、メモリからデータをロードすることなく、キャッシュエントリーに対してタグとしてのアドレスを設定しバリッドフラグをセットし、あるいは、キャッシュエントリーに対してライトバックされていない書き換えられたデータを保持している状態でダーティフラグをリセットするフラグ改変部 4 1 を備える。

【選択図】 図 4

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 4 2 4 0 4 2
受付番号	5 0 3 0 2 1 0 1 9 6 0
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 1 2 月 2 4 日

< 認定情報・付加情報 >

【提出日】 平成15年12月22日



特願 2 0 0 3 - 4 2 4 0 4 2

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 5 8 2 1 ]

1 . 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社